Information Sciences 626 (2023) 1–18

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Feature selection via maximizing inter-class independence and minimizing intra-class redundancy for hierarchical classification

Jie Shi [a,b], Zhengyu Li [a,b], Hong Zhao [a,b,*]

[a] School of Computer Science, Minnan Normal University, Zhangzhou, Fujian 363000, China
[b] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou, Fujian 363000, China

ARTICLE INFO

Article history:
Received 12 May 2022
Received in revised form 3 January 2023
Accepted 5 January 2023
Available online 10 January 2023

Keywords:
Hierarchical feature selection
Hierarchical classification
Structural relation
Feature relation

ABSTRACT

Hierarchical feature selection has proven to be significant in reducing classification difficulty. Many existing hierarchical feature selection methods use the hierarchy in the class space as structural information to improve performance. However, these methods focus on the inter-class structural relations and ignore the intra-class feature correlations, which is challenging as the numbers of classes and features in current datasets are both growing rapidly. In this paper, we propose a hierarchical feature selection method that maximizes inter-class independence and minimizes intra-class redundancy using structure and feature relations. First, we investigate the class hierarchy dependency in the tree structure as the structural relation regularization, which maximizes the independence between unrelated classes in the class structure. Second, we transform the feature correlations into a mathematical representation as feature relation regularization, which minimizes the redundancy in each class on the premise of sparsity. Finally, we unify the two regularization terms into a hierarchical feature selection method to trade off the structural and feature relations. Our method exhibits excellent efficiency and effectiveness compared with other feature selection methods.

1. Introduction

The classification scale has expanded from several to tens of thousands of classes with the improvement of data collection and storage capacity [1]. There are usually hierarchical structures among these classes, such as trees and directed acyclic graphs [2]. Complex data with hierarchical structures exist in many practical applications, such as image data [3–6], and protein data [7,8]. Structural information is beneficial for dividing a large task into a group of relatively small subtasks, which reduces the classification difficulty. In recent years, many research fields have begun to explore classification tasks with hierarchical structures, such as text document [9], recommendation tasks [10], and gene identification [11]. A key problem in these classification tasks is that they contain many repetitive and irrelevant features, which endangers classification performance. Feature selection removes irrelevant and redundant features and is widely used in classification tasks [12]. Traditional feature selection methods assume that all classes are unrelated and select a feature subset to distinguish all classes

* Corresponding author at: School of Computer Science, Minnan Normal University, Zhangzhou, Fujian 363000, China.
E-mail address: hongzhaocn@163.com (H. Zhao).

https://doi.org/10.1016/j.ins.2023.01.048
0020-0255/© 2023 Elsevier Inc. All rights reserved.

[13]. However, traditional feature selection is unsuitable for large-scale classification tasks because it requires a group of shared features to distinguish massive classes.

In recent years, designing hierarchical classification strategies to solve these complex tasks has become a research hotspot [14]. Hierarchical feature selection can leverage hierarchical structures to select different feature sets for different classes, which simplifies classification scale [15]. Wang et al. [16] designed a hierarchical feature selection method based on a random projection technique to remove the redundancy and noise caused by feature randomness. Wang et al. [17] considered the specific and complementary information of different frequency bands for application to hierarchical feature selection, which improves the disease diagnosis accuracy. These methods independently select different features for different classes without considering the class relations. In recent years, hierarchical feature selection methods exploring class dependence relations have been proposed to obtain a compact representation in classification tasks. Zhao et al. [18] presented hierarchical feature selection based on recursive regularization, which fully explores the tree dependence relations. Tuo et al. [19] proposed feature selection based on a subtree graph taking into account the parent and child nodes of each intermediate node simultaneously. These methods make full use of the inter-class dependencies in the hierarchy as structural information to assist different node tasks in selecting accurate features.

However, most hierarchical feature selection methods rarely consider intra-class feature correlations with the rapid growth of features, where the feature correlations include redundancy and sparsity. The feature correlations are essential and complementary for improving performance. Redundancy constraints, such as Pearson coefficient [20] and mutual information [21], select features with low correlations. Sparsity constraints, such as ℓ_1-norm [22] and $\ell_{2,1}$-norm [23,24], select few features to complete the classification under the premise of guaranteeing accuracy. It is a great challenge to integrate the advantages of sparse and redundant constraints to minimize intra-class redundant features in hierarchical structures.

In this paper, we propose a hierarchical feature selection method via maximizing inter-class independence and minimizing intra-class redundancy (HFS-MIMR). Its basic idea is to transform the structure and feature relations into an effective regularization as auxiliary information to improve performance. First, we investigate the hierarchical dependencies in the class space as the structural relation regularization. The sibling relationship is one of the main hierarchical structure dependencies in a tree structure. We model the hierarchical dependency information and expect that the sibling nodes have obvious differences. Second, we represent the feature correlation as the feature relation regularization in mathematical form. Specifically, we construct this regularization directly from measures that can characterize the variable independence and significance. The feature relation regularization combines ℓ_1-norm and ℓ_2-norm to realize high feature sparsity and low feature redundancy. Finally, we establish a hierarchical feature selection method and apply structure and feature relation regularizations. We add certain prior knowledge to our hierarchical method through the two regularizations, which trade off the structure and feature relations.

The proposed method is demonstrated via experiments with protein and image datasets. Evaluation metrics are used to measure the differences between methods in terms of effectiveness and efficiency. Experiments prove that the structure relation regularization enlarges the inter-class distance, while the feature relation regularization reduces intra-class correlations. We compare HFS-MIMR with other hierarchical feature selection methods. The experimental results show that HFS-MIMR is superior to the comparison methods for hierarchical classification tasks.

The remainder of this paper is organized as follows. In Section 2, we review related work. In Section 3, we construct a hierarchical task and present the details of the hierarchical feature selection method. In Section 4, we introduce the experimental setting, including datasets, comparison methods, evaluation measures, and parameter settings. In Section 5, we discuss and analyze the experimental results. Finally, we summarize the HFS-MIMR method and propose further research ideas in Section 6.

2. Related work

In this section, we briefly review the literature on traditional feature selection and hierarchical feature selection.

2.1. Traditional feature selection

Traditional feature selection methods guided by feature constraints have attracted much attention. According to the effect on the features, traditional feature selection is catalogized into low-redundancy and high-sparsity-based feature selection. The former imposes low redundancy constraints on the feature selection process, such as mutual information or Pearson correlation coefficients. Ding et al. [25] and Peng et al. [26] suggested using minimum redundancy and maximum correlation (mRMR) measures in feature selection to reduce redundant features. After this work, Jo et al. [27] improved the mRMR performance, which uses Pearson correlation coefficients as the redundancy measure and *R*-value as the correlation measure. Wang et al. [21] proposed a joint feature selection algorithm based on weights of the improved Relief algorithm and mutual information to select feature subsets with low redundancy and strong classification performance. The above methods reduce the computing load imposed by some highly correlated features.

The latter uses sparse regularization constraints to ensure high feature sparsity. A common regularization term is ℓ_1-norm, which provides a good compromise between reducing complexity and ensuring sparsity. In addition, $\ell_{1/2}$-norm [28] is representative in terms of high sparsity and computational efficiency. Nie et al. [23] proposed a robust feature selection method emphasizing the $\ell_{2,1}$-norm minimization of the loss function and regularization term. Subsequently, $\ell_{2,1}$-norm has been widely used in feature selection [29]. The above methods remove some features without information through sparse regularization, which reduces storage space requirements.

However, the above feature selection methods fail to simultaneously consider redundancy and sparsity, with low redundancy and high sparsity being essential conditions for high-quality feature subsets. The proposed method combines feature redundancy and sparsity in a unified framework to maximize the ability to select discriminant features. Furthermore, these traditional methods assume that classes are independent of each other with no structures, which is not suitable for direct application to increasingly universal classification tasks with hierarchical structures.

2.2. Hierarchical feature selection

The real-world data scales are increasing rapidly, such as the numbers of classes and features. Fortunately, these classes are usually organized in a hierarchy, such as trees and directed acyclic graphs (DAGs) [30]. Hierarchical feature selection combines different hierarchical class relations to improve the feature selection effect. According to the scope of action on the classes, hierarchical feature selection methods are grouped into *global* and *local* classifier-based methods. Global methods directly take the whole class hierarchy to deal with a classification problem. Costa et al. [31] proposed a feature selection method tailored to a global hierarchical classifier combined with a variable neighborhood search (VNS) metaheuristic technique. It is worth mentioning that Cerri et al. [32] constructed a global decision tree classifier called CLUS-HMC to check the quality of the selected features. Based on this work, Lima et al. [33] developed a hybrid feature selection method that combines filtering technology with general VNS, and used CLUS-HMC as a global hierarchical classifier to improve performance.

Local methods train a classifier on each tree node to change the focus of the classification problem from global to local. Secker et al. [34] proposed a feature selection method combined with local hierarchical classification, which selects features from top to bottom. Similarly, Paes et al. [35] combined feature selection with local classifiers of different classes to improve the prediction performance of different hierarchical classifiers. These methods using local classifiers have been optimized using a recursive regularization technique that considers the class hierarchy. Zhao et al. [18] recursively selected appropriate features for each intermediate node, embedding two main tree structure dependencies: parent–child and sibling relationships. Similarly, Tuo et al. [19] proposed feature selection based on a subtree graph, which recursively establishes the subtree of internal nodes to explore the two-way dependency of the current intermediate node in the class space. Huang and Liu [36] tried to explore both the structural information and semantic description of class labels in supervised feature selection. Similarly, the proposed method leverages one kind of main class dependency to distinguish classes with sibling relationships to the greatest extent.

However, unlike the above work, the proposed method explores both the class structure and feature information to jointly mine the relationships between different classes within hierarchical structures and the feature relationships in each class.

3. Proposed feature selection method

In this section, we first present an overview of the proposed hierarchical feature selection method via maximizing inter-class independence and minimizing intra-class redundancy (HFS-MIMR). Then, we introduce each part of HFS-MIMR in detail. Finally, we optimize the HFS-MIMR solution process.

3.1. Overview of the HFS-MIMR method

HFS-MIMR consists of four terms, including a loss constraint and three regularization constraints, as shown in Fig. 1. The constraints include the sparse, structural, and feature relation regularization constraints. (1) **Loss**: We use the least-squares

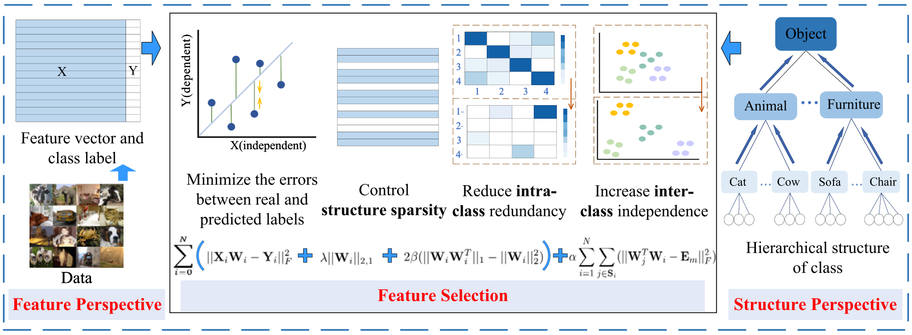

X
Y
Feature vector and class label
Data
Feature Perspective
Y(dependent)
X(independent)
1
2
3
4
Minimize the errors between real and predicted labels
Control structure sparsity
Reduce intra-class redundancy
Increase inter-class independence
$\sum_{i=0}^{N}\left(\|\mathbf{X}_i\mathbf{W}_i-\mathbf{Y}_i\|_F^2+\lambda\|\mathbf{W}_i\|_{2,1}+2\beta(\|\mathbf{W}_i\mathbf{W}_i^T\|_1-\|\mathbf{W}_i\|_2^2)\right)+\alpha\sum_{i=1}^{N}\sum_{j\in\mathbf{S}_i}(\|\mathbf{W}_j^T\mathbf{W}_i-\mathbf{E}_m\|_F^2)$
Feature Selection
Object
Animal
Furniture
Cat
Cow
Sofa
Chair
Hierarchical structure of class
Structure Perspective

Fig. 1. Framework of the HFS-MIMR method.

method as a loss constraint to minimize the errors between the real and predicted labels. (2) **Sparse regularization**: Sparse regularization controls the structure sparsity to quickly select the optimal shared features. (3) **Structural relation regularization**: Inter-class independence is increased by the structural relation constraint. (4) **Feature relation regularization**: Intra-class redundancy is reduced by the feature relation constraint. We apply one loss and three relation constraints in the hierarchical feature selection process. The notations for variables, matrices, formulas, and corresponding explanations are listed in Table 1.

3.2. A basic hierarchical method based on sparse learning

A complex classification task is decomposed into a group of sub-tasks with the help of a tree structure. The tree nodes are divided into leaf and non-leaf nodes, where non-leaf nodes include a root node and intermediate nodes. We correspond the sub-tasks to the non-leaf nodes and use Example 1 to illustrate the process of the decomposition tasks. We effectively focus on the current sub-task to select a feature subset for each sub-task.

Example 1. Fig. 2 shows the top-down decomposition process of the hierarchical tasks. For the root node task "*Object*", we focus on whether the sample belongs to the coarse classes "*Animal*" or "*Furniture*". For the intermediate node task "*Furniture*", we distinguish whether the sample belongs to "*Chair*" or "*Sofa*". Intermediate node tasks are further classified until the fine class of the sample is determined.

We introduce a basic hierarchical feature selection method. The goal of hierarchical feature selection is to jointly minimize the fitting error and the sparse regularization. Sparse learning is combined with hierarchical feature selection to improve the algorithm prediction accuracy due to its good interpretability [37]. Let $\mathbf{Y}_i \in \mathbf{R}^{n_i \times m}$ be a class matrix of the i^{th} non-leaf node, and $\mathbf{Y}_i = [\mathbf{y}^i_1; \mathbf{y}^i_2; \cdots; \mathbf{y}^i_{n_i}]$. Let $\mathbf{X}_i \in \mathbf{R}^{n_i \times d}$ be a data matrix of the i^{th} non-leaf node, and $\mathbf{X}_i = [\mathbf{x}^i_1; \mathbf{x}^i_2; \cdots; \mathbf{x}^i_{n_i}]$. The hierarchical feature selection method based on sparse learning is defined as follows

$$\mathbf{J} = \min_{\mathbf{W}_0, \mathbf{W}_1, \cdots, \mathbf{W}_N} \sum_{i=0}^{N} (L(\mathbf{X}_i, \mathbf{W}_i, \mathbf{Y}_i) + \lambda S(\mathbf{W}_i)), \tag{1}$$

where $L(\cdot)$ is a loss that measures the error between the real and predicted labels and $S(\cdot)$ represents a sparse regularization. Parameter λ is a positive parameter that controls the sparsity, and N is the non-leaf node number. Matrix $\mathbf{W}_i \in \mathbf{R}^{d \times m}$ represents the weight of the i^{th} non-leaf node, and $\mathbf{W}_i = [\mathbf{w}^i_1; \mathbf{w}^i_2; \cdots; \mathbf{w}^i_j; \cdots; \mathbf{w}^i_{m_i}]$.

According to [38], the loss can be divided into the least-squares, hinge, empirical, and logistic losses. The least-square loss transforms a regression problem into a convex optimization problem, which is used in linear regression. It has the following advantages: (1) the least-squares loss uses the Euclidean distance as a similarity measure, which is simple to calculate; and (2) the feature attributes remain unchanged after transforming different representation domains. Therefore, the least-squares loss is applied to hierarchical feature selection as follows

$$L(\mathbf{X}_i, \mathbf{W}_i, \mathbf{Y}_i) = \sum_{i=0}^{N} ||\mathbf{X}_i \mathbf{W}_i - \mathbf{Y}_i||_F^2, \tag{2}$$

where $|| \cdot ||_F^2$ represents the Frobenius norm of a matrix.

From the perspective of the constraint structure, sparse regularizations are divided into two categories: (1) regularization for flat sparsity, which is suitable for binary classification, and (2) regularization for structure sparsity, which is suitable for multi-classification. We adopt structure sparsity regularization to solve hierarchical classification tasks. The terms $\ell_{2,0}$-norm and $\ell_{2,1}$-norm are usually used for structural sparse regularization. The $\ell_{2,0}$-norm defines the number of selected features in advance and easily falls into a local optimum. In contrast, the $\ell_{2,1}$-norm reduces the feature weight to near zero. Meanwhile,

Table 1
Description of symbols in the proposed method. Matrices are represented in bold capital letters and variables are represented in lowercase letters.

Notation	Representation
n_i	The sample number in sub-tree of i^{th} non-leaf node
m	The maximum class number in all sub-trees
d	The feature number
$\mathbf{x}^i_j$	The j^{th} sample matrix at the i^{th} non-leaf node, and $\mathbf{x}^i_j \in \mathbf{R}^{d \times 1}$
$\mathbf{w}^i_j$	The j^{th} feature weight at the i^{th} non-leaf node, and $\mathbf{w}^i_j \in \mathbf{R}^{d \times 1}$
$\mathbf{y}^i_j$	The j^{th} class matrix at the i^{th} non-leaf node, and $\mathbf{y}^i_j \in \{0,1\}^{m \times 1}$
$\mathbf{E}_m$	The m-by-m unit matrix
$\mathbf{S}_i$	The sibling node set of i^{th} non-leaf node

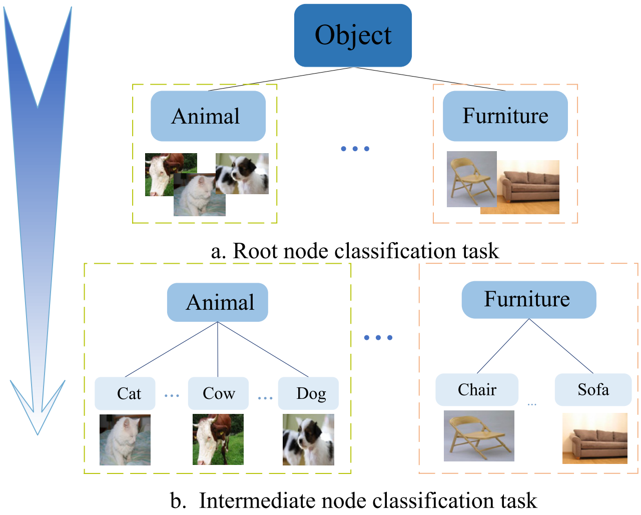

Object
Animal
Furniture
a. Root node classification task
Animal
Cat
Cow
Dog
Furniture
Chair
Sofa
b. Intermediate node classification task

Fig. 2. Top-down decomposition process used in a hierarchical classification task.

the $\ell_{2,1}$-norm is convex and globally optimized. Therefore, we take the $\ell_{2,1}$-norm as the sparse regularization of hierarchical feature selection. The $\ell_{2,1}$-norm of $\mathbf{W}_i$ is defined as: $||\mathbf{W}_i||_{2,1} = \sum_{j=1}^{d}||\mathbf{w}_j^i||_2$.

We combine the least-squares loss and $\ell_{2,1}$-norm sparse regularization into a unified method. The primary optimization problem in hierarchical feature selection based on sparse learning (HFS-Sparse) is to minimize $J_{spar}(\mathbf{W}_0, \mathbf{W}_i, \cdots, \mathbf{W}_N)$:

$$J_{spar}(\mathbf{W}_0, \mathbf{W}_i, \cdots, \mathbf{W}_N) = \sum_{i=0}^{N}(||\mathbf{X}_i\mathbf{W}_i - \mathbf{Y}_i||_F^2 + \lambda||\mathbf{W}_i||_{2,1}). \tag{3}$$

HFS-Sparse retains important features and has good interpretability for sparse features.

3.3. A hierarchical method based on structural relations

A hierarchical structure, as prior knowledge, is conducive to exploring the dependencies between classes to capture the inherent information in the class space. Sibling relationships are one of the main structural dependencies and consider inter-class relationships in hierarchical tree structures. Sibling classes are the closest heterogeneous classes with a common parent class, and the features identifying sibling classes need to be significantly distinctive [36]. We use the hierarchical structure example of "*Object*" as shown in Fig. 3 to illustrate the dependency relationship in the tree structure.

From Fig. 3, we obtain the following conclusions. (1) The intermediate classes "*Cat*", "*Cow*", and "*Dog*" have the same parent class "*Animal*" and have their sub-classes that are sibling classes. (2) The sibling classes not only have common features but also unique ones. To classify these sub-classes, these sibling classes each need to conduct feature selection respectively.

The feature space selected by the HFS-Sparse method includes invalid and discriminant features, where the invalid features unrelated to classification appear in the private feature spaces of the sibling classes [39]. Orthogonal constraints encourage the selection of unique features for sibling classes to reduce the mutual interference between special feature

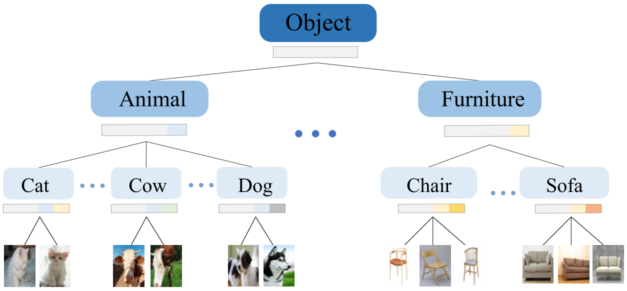


Fig. 3. A hierarchical structure of "Object". The common features are represented in a common color. Different colors distinguish the unique features of the sibling nodes.

spaces [39]. We design the structural relation regularization $R_1(\mathbf{W}_i, \mathbf{W}_j)$ based on orthogonal constraints to constrain the similarity between sibling classes, which is defined as

$$R_1(\mathbf{W}_i, \mathbf{W}_j) = \sum_{j \in \mathbf{S}_i} ||\mathbf{W}_j^T \mathbf{W}_i - \mathbf{E}_m||_F^2, \tag{4}$$

where $j \in \mathbf{S}_i$ represents the j^{th} sibling node of the i^{th} non-leaf node, and $\mathbf{E}_m$ is a unit matrix. The structural relation regularization requires that the selected feature subsets between sibling classes are independent of each other, which encourages inter-class independency.

Based on the above discussion, the primary optimization problem when considering class structure dependency is to minimize $J_{stru}(\mathbf{W}_0, \mathbf{W}_i, \cdots, \mathbf{W}_N)$:

$$\begin{aligned} J_{stru}(\mathbf{W}_0, \mathbf{W}_i, \cdots, \mathbf{W}_N) = \; & \textstyle\sum_{i=0}^{N} (||\mathbf{X}_i \mathbf{W}_i - \mathbf{Y}_i||_F^2 + \lambda ||\mathbf{W}_i||_{2,1} \\ & + \alpha \sum_{i=1}^{N} \sum_{j \in \mathbf{S}_i} (||\mathbf{W}_j^T \mathbf{W}_i - \mathbf{E}_m||_F^2), \end{aligned} \tag{5}$$

where α is a parameter that balances the magnitudes between the sibling relations in the tree structure. This task selects features to improve inter-class independence and is called HFS-Inter.

3.4. A hierarchical method based on feature relations

Feature relations, as auxiliary knowledge, can improve the quality of the feature space and reduce the negative impact of redundant features. For any node in the tree structure, the features are independent and distinct from each other. Inspired by [40], we introduce the absolute values of the eigenvector inner product to directly measure feature independence and discrimination. The inner product term has the following properties. (1) The non-informative feature weight in $\mathbf{W}_i$ is zero, which reserves the feature subset most relevant to classification. (2) The correlation between rows in $\mathbf{W}_i$ represents the correlation between features. We select the feature subset with the lowest correlation between features. The small inner product term greatly reduces the weighting of invalid and redundant features. Based on these aspects, the feature relation regularization $R_2(\mathbf{W}_i)$ that discards redundant and unimportant features is defined as

$$\begin{aligned} R_2(\mathbf{W}_i) & = \sum_{j=1}^{d} \sum_{k=1, k \neq j}^{d} | < \mathbf{w}_j^i, \mathbf{w}_k^i > | \\ & = \sum_{j=1}^{d} \sum_{k=1}^{d} | < \mathbf{w}_j^i, \mathbf{w}_k^i > | - \sum_{j=1}^{d} | < \mathbf{w}_j^i, \mathbf{w}_j^i > |, \end{aligned} \tag{6}$$

where $\mathbf{w}_j^i$ and $\mathbf{w}_k^i$ are the weight vectors of the j^{th} and k^{th} rows, respectively. By simple algebra transformations, the feature relation regularization is expressed in the joint form of ℓ_1-norm and ℓ_2-norm:

$$\begin{aligned} R_2(\mathbf{W}_i) & = (||\mathbf{W}_i \mathbf{W}_i^T||_1 - Tr(\mathbf{W}_i^T \mathbf{W}_i)) \\ & = (||\mathbf{W}_i \mathbf{W}_i^T||_1 - ||\mathbf{W}_i||_2^2), \end{aligned} \tag{7}$$

where $|| \cdot ||_1$ represents the ℓ_1-norm, and $|| \cdot ||_2^2$ represents the ℓ_2-norm.

We embed feature relation into HFS-Sparse with feature high sparsity and low redundancy. The primary optimization problem when considering feature relations is to minimize $J_{feat}(\mathbf{W}_0, \mathbf{W}_i, \cdots, \mathbf{W}_N)$:

$$\begin{aligned} J_{feat}(\mathbf{W}_0, \mathbf{W}_i, \cdots, \mathbf{W}_N) & = \sum_{i=0}^{N} (||\mathbf{X}_i \mathbf{W}_i - \mathbf{Y}_i||_F^2 + \lambda ||\mathbf{W}_i||_{2,1} \\ & + 2\beta (||\mathbf{W}_i \mathbf{W}_i^T||_1 - ||\mathbf{W}_i||_2^2)), \end{aligned} \tag{8}$$

where β is a parameter that punishes the redundant feature degree.

This task selects features to reduce intra-class redundancy and is called HFS-Intra.

3.5. A joint hierarchical method based on structural and feature relations

We embed two external knowledge, structural and feature relations, into HFS-Sparse. The primary optimization problem when constraining these two relations jointly is to minimize $J(\mathbf{W}_0, \mathbf{W}_i, \cdots, \mathbf{W}_N)$:

$$\begin{aligned} J(\mathbf{W}_0, \mathbf{W}_i, \cdots, \mathbf{W}_N) = & \sum_{i=0}^{N} (||\mathbf{X}_i\mathbf{W}_i - \mathbf{Y}_i||_F^2 + \lambda||\mathbf{W}_i||_{2,1} \\ & +2\beta(||\mathbf{W}_i\mathbf{W}_i^T||_1 - ||\mathbf{W}_i||_2^2)) + \alpha\sum_{i=1}^{N}\sum_{j\in\mathbf{S}_i}(||\mathbf{W}_j^T\mathbf{W}_i - \mathbf{E}_m||_F^2). \end{aligned} \tag{9}$$

This task selects features by maximizing inter-class independence and minimizing intra-class redundancy and is called HFS-MIMR. By optimizing Eq. (9), we obtain the feature weight matrix $\mathbf{W}_i$ of the i^{th} non-leaf node. We sort the feature weights in descending order in the corresponding task to select the top-ranked features as the optimal features.

Compared with previous hierarchical feature selection methods, HFS-MIMR has the following merits. First, we introduce class dependencies to constrain the feature selection process. Second, feature relations are considered to select features with low correlation. HFS-MIMR reduces intra-class redundancy and increases inter-class independence.

3.6. Optimization algorithm

We introduce the optimization algorithm of the proposed method HFS-MIMR in detail. All nodes are divided into two categories in the tree structure: a root node without siblings and intermediate nodes with siblings. Therefore, the final objective function of Eq. (9) is reformulated as

$$\begin{aligned} J(\mathbf{W}_0, \mathbf{W}_1, \cdots, \mathbf{W}_N) = & (||\mathbf{X}_0\mathbf{W}_0 - \mathbf{Y}_0||_F^2 + \lambda||\mathbf{W}_0||_{2,1} \\ & +2\beta(||\mathbf{W}_0\mathbf{W}_0^T||_1 - ||\mathbf{W}_0||_2^2)) + \sum_{i=1}^{N}(||\mathbf{X}_i\mathbf{W}_i - \mathbf{Y}_i||_F^2 + \lambda||\mathbf{W}_i||_{2,1} \\ & +\alpha\sum_{j\in\mathbf{S}_i}||\mathbf{W}_j^T\mathbf{W}_i - \mathbf{E}_m||_F^2 + 2\beta(||\mathbf{W}_i\mathbf{W}_i^T||_1 - ||\mathbf{W}_i||_2^2)). \end{aligned} \tag{10}$$

We optimize and analyze the different cases of Eq. (10).

In one case, HFS-MIMR is applied to the root node without siblings expressed as

$$\begin{aligned} J(\mathbf{W}_0) \quad & = ||\mathbf{X}_0\mathbf{W}_0 - \mathbf{Y}_0||_F^2 + \lambda||\mathbf{W}_0||_{2,1} \\ & +2\beta(||\mathbf{W}_0\mathbf{W}_0^T||_1 - ||\mathbf{W}_0||_F^2). \end{aligned} \tag{11}$$

After deformation, the objective function for the root is equivalent to

$$\begin{aligned} J(\mathbf{W}_0) = \quad & ||\mathbf{X}_0\mathbf{W}_0 - \mathbf{Y}_0||_F^2 + \lambda Tr(\mathbf{W}_0^T\mathbf{D}_0\mathbf{W}_0) \\ & +2\beta Tr(\mathbf{1}_{d\times d}\mathbf{W}_0\mathbf{W}_0^T - \mathbf{W}_0\mathbf{W}_0^T). \end{aligned} \tag{12}$$

The $\ell_{2,1}$-norm non-smoothness makes optimization of the equation difficult. Reference [41] provides the $\ell_{2,1}$-norm derivation method:

$$\frac{\partial(||\mathbf{W}||_{2,1})}{\partial\mathbf{W}} = \frac{\partial(Tr(\mathbf{W}^T\mathbf{D}\mathbf{W}))}{\partial\mathbf{W}} = 2\mathbf{D}\mathbf{W}, \tag{13}$$

where $\mathbf{D} \in \mathbf{R}^{d\times d}$ is a diagonal matrix with the i^{th} diagonal element being $D_{ii} = \frac{1}{2||\mathbf{W}_i^j||_2}$.

By setting the derivative of $\mathbf{W}_0$ in Eq. (12) to zero, we have

$$\begin{aligned} \tfrac{\partial J}{\partial \mathbf{W}_0} \quad & = 2\mathbf{X}_0^T(\mathbf{X}_0\mathbf{W}_0 - \mathbf{Y}_0) + 2\lambda\mathbf{D}_0\mathbf{W}_0 + 2\beta(\mathbf{1}_{m\times m} - \mathbf{E}_m)\mathbf{W}_0 \\ & = 2(\mathbf{X}_0^T\mathbf{X}_0 + \lambda\mathbf{D}_0 + \beta\mathbf{1}_{m\times m} - \beta\mathbf{E}_m)\mathbf{W}_0 - 2\mathbf{X}_0^T\mathbf{Y}_0. \end{aligned} \tag{14}$$

Therefore, we obtain $\mathbf{W}_0$ as

$$\mathbf{W}_0 = (\mathbf{X}_0^T\mathbf{X}_0 + \lambda\mathbf{D}_0 + \beta\mathbf{1}_{m\times m} - \beta\mathbf{E}_m)^{-1}(\mathbf{X}_0^T\mathbf{Y}_0). \tag{15}$$

In another case, HFS-MIMR is applied to the intermediate nodes with siblings. The objective function for the intermediate nodes is formulated as

$$\begin{aligned} J(\mathbf{W}_1, \cdots, \mathbf{W}_N) = & \sum_{i=1}^{N}(||\mathbf{X}_i\mathbf{W}_i - \mathbf{Y}_i||_F^2 + \lambda||\mathbf{W}_i||_{2,1} \\ & +\alpha\sum_{j\in\mathbf{S}_i}||\mathbf{W}_j^T\mathbf{W}_i - \mathbf{E}_m||_F^2 + 2\beta(||\mathbf{W}_i\mathbf{W}_i^T||_1 - ||\mathbf{W}_i||_2^2)). \end{aligned} \tag{16}$$

After a series of linear transformations, Eq. (16) is equivalent to

$$\begin{aligned} J(\mathbf{W}_1,\cdots,\mathbf{W}_N) = &\sum_{i=1}^{N}(||\mathbf{X}_i\mathbf{W}_i-\mathbf{Y}_i||_F^2+\lambda Tr(\mathbf{W}_i^T\mathbf{D}_i\mathbf{W}_i) \\ &+\alpha\sum_{j\in\mathbf{S}_i}||\mathbf{W}_j^T\mathbf{W}_i-\mathbf{E}_m||_F^2+2\beta Tr(\mathbf{1}_{m\times m}\mathbf{W}_i\mathbf{W}_i^T-\mathbf{W}_i\mathbf{W}_i^T)). \end{aligned} \tag{17}$$

By taking the derivative of the $\mathbf{W}_i$ in Eq. (17), it is expressed as

$$\begin{aligned} \tfrac{\partial J}{\partial \mathbf{W}_i} &= 2\mathbf{X}_i^T(\mathbf{X}_i\mathbf{W}_i-\mathbf{Y}_i)+2\lambda\mathbf{D}_i\mathbf{W}_i \\ &+2\alpha\sum_{j\in\mathbf{S}_i}\mathbf{W}_j(\mathbf{W}_j^T\mathbf{W}_i-\mathbf{E}_m)+2\beta(\mathbf{1}_{m\times m}-\mathbf{E}_m)\mathbf{W}_i \\ &=2(\mathbf{X}_i^T\mathbf{X}_i+\lambda\mathbf{D}_i+\alpha\sum_{j\in\mathbf{S}_i}\mathbf{W}_j\mathbf{W}_j^T+\beta\mathbf{1}_{m\times m}-\beta\mathbf{E}_m)\mathbf{W}_i \\ &-2(\mathbf{X}_i^T\mathbf{Y}_i+\alpha\sum_{j\in\mathbf{S}_i}\mathbf{W}_j\mathbf{E}_m). \end{aligned} \tag{18}$$

By setting the value in Eq. (18) to zero, we obtain

$$\begin{aligned} \mathbf{W}_i = \ &(\mathbf{X}_i^T\mathbf{X}_i+\lambda\mathbf{D}_i+\alpha\sum_{j\in\mathbf{S}_i}\mathbf{W}_j\mathbf{W}_j^T+\beta\mathbf{1}_{m\times m}-\beta\mathbf{E}_m)^{-1} \\ &(\mathbf{X}_i^T\mathbf{Y}_i+\alpha\sum_{j\in\mathbf{S}_i}\mathbf{W}_j\mathbf{E}_m). \end{aligned} \tag{19}$$

We present the HFS-MIMR algorithm in Algorithm 1 and explain it briefly. The updated calculations of the non-leaf nodes are in lines 3 to 13. The diagonal matrix is calculated in lines 4 to 6. The update of the root node $\mathbf{W}_0$ is in line 7. The update of intermediate nodes $\mathbf{W}_i$ needs to be recursive from lines 8 to 10. We obtain that the HFS-MIMR time complexity largely depends on the update time of the non-leaf nodes $\mathbf{W}$. The time complexity of the root nodes is calculated as $O(d^3+d^2m)$. The time complexity of the intermediate nodes is expressed as $d^2n_i+dn_im$, where n_i is the sample number of the i^{th} intermediate node. The $\mathbf{X}_i^T\mathbf{X}_i$ and $\mathbf{X}_i^T\mathbf{Y}_i$ calculations of each intermediate node require d^2n_i and dn_im operations, respectively. The total update of all intermediate nodes requires d^2n and dnm operations. Therefore, the HFS-MIMR time complexity is expressed as $O(N(d^3+d^2m)+d^2n+dnm)$, where N is the total iteration number.

Algorithm 1: The algorithm of HFS-MIMR.

```
Input: (1) Feature matrix $\mathbf{X}_i \in \mathbf{R}^{n_i\times d}$; (2) Label matrix $\mathbf{Y}_i \in \mathbf{R}^{n_i\times m}$; (3) Regularization parameters: $\lambda$, $\beta$, and $\alpha$; (4)
  Parameter $i = 0,1,\cdots,N$ and $N$ indicates the non-leaf node number; and (5) The maximal iteration number $T$.
Output: Weight matrix $\mathbf{W} \in \mathbf{R}^{d\times m}$.
1. Set iteration number $t = 0$ and initialize $\mathbf{W} \in \mathbf{R}^{n\times d}$ randomly;
2. $\mathbf{W} = [\mathbf{W}_0,\mathbf{W}_1,\cdots,\mathbf{W}_N]$;
3. while $t < T$ do
4.   for $i = 0 : N$ do
5.     Compute the diagonal matrix $\mathbf{D}_i^{(t)}$ according to Eq. (13);
6.   end for
7.   Update $\mathbf{W}_0$: $\mathbf{W}_0^{(t+1)} = (\mathbf{X}_0^T\mathbf{X}_0+\lambda\mathbf{D}_0^{(t)}+\beta\mathbf{1}_{m\times m}-\beta\mathbf{E}_m)^{-1}(\mathbf{X}_0^T\mathbf{Y}_0)$;
8.   for $i = 1 : N$ do
9.     Update $\mathbf{W}_i$: $\mathbf{W}_i^{(t+1)} = (\mathbf{X}_i^T\mathbf{X}_i+\lambda\mathbf{D}_i^{(t)}+\beta\mathbf{1}_{m\times m}-\beta\mathbf{E}_m+\alpha\sum_{j\in\mathbf{S}_i}\mathbf{W}_j\mathbf{W}_j^T)^{-1}(\mathbf{X}_i^T\mathbf{Y}_i+\alpha\sum_{j\in\mathbf{S}_i}\mathbf{W}_j\mathbf{E}_m)$;
10.   end for
11.   Update $\mathbf{W}^{(t+1)} = [\mathbf{W}_0^{(t+1)},\mathbf{W}_1^{(t+1)},\cdots,\mathbf{W}_N^{(t+1)}]$;
12.   $t = t + 1$;
13. end while
14. return $\mathbf{W}$;
```

4. Experimental setting

In this section, we describe the experimental setting in terms of the (1) experimental datasets, (2) comparison methods, (3) evaluation indicators, and (4) experimental parameter settings.

4.1. Experimental dataset descriptions

Six public datasets with hierarchical structures, including protein and image datasets, are used in the experiments. These datasets are single-label, and all samples are allocated to leaf nodes. The introductions of these datasets are listed in Table 2.

The protein datasets include *F194* [8] and *DD* [7], where (1) *F194* deletes samples that rarely appear in the class to become a dataset containing 7,105 samples, and (2) *DD* divides 27 classes into 4 coarse classes. The image datasets include *ILSVRC65* [3], *VOC* [4], *CIFAR100* [5], and *SUN* [6], which are described as follows. (1) The *ILSVRC65* dataset is managed by the WordNet hierarchy. (2) *VOC* is a benchmark dataset used in visual class recognition and detection. (3) *Cifar100* consists of 100 fine classes, which are divided into 20 coarse classes. Each class has fine and coarse class labels. (4) *SUN* deletes the classes containing multi-labels to obtain a dataset containing 324 classes.

4.2. Comparison methods

In this section, we compare several hierarchical methods, including classical and advanced methods, to evaluate the performance of the HFS-MIMR method.

The classical methods include HFisher, HFSNM, and HMRMR, which are described below. HFisher applies the Fisher [42] idea to the hierarchal structure, which selects features according to the distances between the labeled data. HFSNM jointly minimizes the $\ell_{2,1}$-norm of the loss and regularization terms to select features by using FSNM [23] on hierarchal tasks. HMRMR [43] applies the minimum redundancy and maximum correlation criteria to the hierarchical feature selection process.

The advanced methods include HiRRfam [44], HFSDK [45], HFSLDL [46], and LCCSHFS [47], which are presented below. HiRRfam fully considers the class relationship in the hierarchy and uses recursive regularization technology in the feature selection process. HFSDK adds semantic constraints to the hierarchical class structure and eliminates data anomalies by the upper bound hinge loss. HFSLDL evaluates the similarities between different classes in hierarchical tasks using label enhancement to select identification features. LCCSHFS explores the specific and common features of different nodes in the hierarchical structure to select the feature subset most relevant to classification tasks.

4.3. Evaluation indicators

We assess the performance based on evaluation indicators divided into flat and hierarchical indicators. We adopt hierarchical evaluation indicators of two categories: *efficiency* and *effectiveness*.

The efficiency evaluation indicators include R-Time and T-Time, which are described in detail below. (1) R-Time is the running time used to select features, which evaluates operation efficiency. (2) T-Time is the classifier test time used for evaluating the selected features.

The effectiveness evaluation indicators include ACC, TIE, Hier-F_1, and F_{LCA}, which are described below. (1) ACC is a simple flat evaluation method that evaluates the proportion of predicted correct samples. Unlike ACC, the following three evaluation indicators are unique to hierarchical methods. (2) TIE [48] measures the errors caused by the hierarchical class structure while considering the relationship between the real and predicted classes. (3) Hier-F_1 [49] is a set-based evaluation that considers both the ancestors and descendants of the real and predicted classes. (4) F_{LCA} [50] is an ancestor assessment based on graph theory, which eliminates the impact of having too many ancestor nodes.

4.4. Parameter settings

We set the experimental parameters λ, α, and β in advance. (1) The parameter λ controls sparsity. A high λ value causes some important features to be ignored, while a small λ value fails to reduce unimportant feature weights. (2) The parameter α adjusts the tightness between classes in the hierarchical structures. A high α value causes common features in sibling classes to be neglected, which is detrimental to classification. A small α value makes it difficult to distinguish the differences between sibling classes. (3) The parameter β punishes the redundant features in each class. A small β value results in ineffective avoidance of the effects of redundant features. A high β value suppresses the influence of some features related to classification. Therefore, the parameter settings used with different types of datasets should be distinguished.

Table 2
Descriptions of six experimental datasets with hierarchical structure.

Dataset	Domain	Feature	Height	Training	Test	Class
DD	Protein	473	3	3,020	605	27
F194	Protein	473	3	7,105	1,420	194
VOC	Image	4,096	3	3,437	3,539	20
Cifar100	Image	4,096	3	50,000	10,000	100
ILSVRC65	Image	4,096	4	12,346	11,845	57
SUN	Image	4,096	4	45,109	22,556	324

The parameter settings are as follows. The parameter λ is set to 10 on the protein datasets and 100 on the image datasets. The parameters α and β are tuned in the set {0.01, 0.1, 1, 10, 100}. The optimal parameters are $\alpha = 0.1, \beta = 0.1$ on the protein datasets and $\alpha = 1, \beta = 1$ on the image datasets.

We select 10% and 20% of all features from the protein and image datasets, respectively. A top-down linear support vector machine (SVM) classifier for 10-fold cross-validation is used to evaluate the selected features. All experiments are completed on a desktop computer equipped with an Intel Core i7-3770, 3.40 GHz CPU, 32.0 GB memory, and the 64-bit Windows 10 operating system.

5. Experimental results and analysis

In this section, the proposed method is evaluated from the following aspects: (1) performance comparison of different methods; (2) efficiency comparison with other methods; (3) performance comparison using different selected feature numbers; (4) visualization of structural and feature relation regularization terms; and (5) parameter sensitivity analysis. Many datasets contain tens of thousands of samples and thousands of features, such as *Cifar100* and *SUN*. Some feature selection methods consume a lot of memory space in processing such datasets, resulting in insufficient memory. Therefore, we use "—" to represent this situation. The best results in the tables are highlighted in bold.

5.1. Performance comparison of different methods

In this subsection, we analyze the performance of different feature selection methods using several evaluation indicators. These indicators are divided into two categories: (1) the flat indicator ACC, and (2) the hierarchical evaluation indicators TIE, Hier-F_1, and F_{LCA}. The symbol "↑" means larger values are better, and "↓" means smaller values are better.

Table 3 shows the normalized TIE values using different methods on the experimental dataset.

From Table 3, we obtain the following observations:

(1) The TIE value represents the error edge number and is related to the class size in the hierarchy. For example, *SUN* and *VOC* are image datasets with 20 and 324 classes, respectively. The TIE values of *SUN* are greater than those of *VOC*. Thus, we obtain that a class increment leads to a TIE value increment.

(2) HFS-MIMR is more effective on most datasets than other comparison methods. It has a better ability to select distinguishing features than classical algorithms. For example, the TIE value of HFS-MIMR is 0.033 greater than the HFisher value on the *SUN* dataset. Compared with advanced algorithms, HFS-MIMR exhibits great advantages, especially in its ability to process large-scale datasets. For example, the memory requirements of the HFSLDL and HFSDK algorithms are too high for them to handle large datasets. HFS-MIMR has advantages in processing different types of datasets, such as image and protein datasets. For example, the TIE values of LCCSHFS on all datasets are inferior to those of HFS-MIMR.

Table 4 shows the Hier-F_1, F_{LCA}, and ACC values using different feature selection methods on different datasets.

From Table 4, we obtain the following conclusions:

(1) The Hier-F_1 value is higher than the F_{LCA} value, which indicates that too many common ancestors reduce the experimental effect.

(2) HFS-MIMR has good performance in selecting identification features for classification tasks. Compared with classical algorithms, HFS-MIMR has better classification effectiveness. For example, the Hier-F_1 value of HFS-MIMR is 11.95 higher than that of HFisher on the *DD* dataset. Compared with advanced algorithms, HFS-MIMR is inferior to HiRRfam only on the *SUN* dataset. Due to insufficient computer memory, many methods fail to run on datasets containing many classes, such as HFSLDL and HFSDK.

Table 3
TIE values of different feature selection methods on different datasets (↓).

Method	*F194*	*DD*	*VOC*	*Cifar100*	*SUN*	*ILSVRC65*
HFisher	1.945	1.355	1.160	1.285	1.341	0.336
HFSNM	2.123	0.886	1.191	–	–	0.35
HMRMR	1.800	0.919	1.155	1.273	1.322	0.335
HiRRfam	1.730	0.85	1.154	1.272	1.271	0.329
HFSDK	1.750	0.84	1.171	–	–	0.334
HFSLDL	1.704	0.836	1.204	–	–	0.346
LCCSHFS	1.708	0.833	1.222	1.402	1.330	0.368
HFS-MIMR	1.700	0.826	0.272	1.249	1.308	0.321

Table 4
The Hier-F_1, F_{LCA} and ACC values of different feature selection methods on different datasets (↑).

Method	Metric	Datasets					
		F194	*DD*	*VOC*	*Cifar100*	*SUN*	*ILSVRC65*
HFisher	Hier-F_1	63.66	74.17	79.81	78.59	83.24	95.80
	F_{LCA}	59.47	73.80	76.48	76.42	76.33	92.23
	ACC	25.77	52.07	58.86	61.38	61.18	85.01
HFSNM	Hier-F_1	60.23	76.31	79.34	–	–	95.63
	F_{LCA}	57.37	81.58	75.95	–	–	91.88
	ACC	24.51	66.96	58.01	–	–	84.31
HMRMR	Hier-F_1	63.45	77.47	79.87	78.79	83.48	95.81
	F_{LCA}	62.50	82.30	76.50	76.65	76.69	92.22
	ACC	32.11	68.28	58.63	61.78	61.77	84.97
HiRRfam	Hier-F_1	71.62	85.96	80.15	78.78	**84.10**	95.87
	F_{LCA}	63.88	82.52	76.69	76.61	**77.40**	92.26
	ACC	34.23	68.61	59.06	61.73	**62.79**	84.97
HFSDK	Hier-F_1	70.92	85.90	79.65	–	–	95.83
	F_{LCA}	63.30	82.54	76.26	–	–	92.27
	ACC	33.52	68.77	58.46	–	–	85.09
HFSLDL	Hier-F_1	71.60	86.07	79.16	–	–	95.51
	F_{LCA}	63.78	82.57	75.45	–	–	91.61
	ACC	33.94	68.61	56.85	–	–	83.76
LCCSHFS	Hier-F_1	71.29	85.90	78.88	76.73	83.35	95.40
	F_{LCA}	63.67	82.54	75.47	74.55	76.44	91.42
	ACC	34.08	68.77	57.14	58.56	61.30	83.39
HFS-MIMR	Hier-F_1	**71.71**	**86.12**	**80.34**	**79.18**	83.65	**96.01**
	F_{LCA}	**64.07**	**82.65**	**76.86**	**77.09**	76.91	**92.54**
	ACC	**34.65**	**68.78**	**59.40**	**62.49**	62.09	**85.54**

5.2. Efficiency comparison with other methods

In this subsection, we use R-time and T-Time to compare the time efficiency of different methods, including the operational and test efficiencies.

First, we use R-time to measure the operational efficiency of different methods. Table 5 shows the R-time values of different feature selection methods.

From Table 5, we obtain the following conclusions:

(1) The processing efficiency of HFS-MIMR is better than that of most advanced algorithms, especially on large-scale datasets. For example, the time efficiency of HFS-MIMR is one order of magnitude better than that of LCCSHFS on the *SUN* dataset. Many methods are unable to select valuable features on large-scale datasets, such as HFSLDL and HFSDK.
(2) The operation efficiency of HFS-MIMR is superior to most classical algorithms, except HFisher. However, the results in Tables 3 and 4 show that HFS-MIMR performs better than HFisher in terms of effectiveness. This is the tradeoff between efficiency and effectiveness. Compared with HFisher, HFS-MIMR selects good effectiveness.

Next, we compare the test efficiency in terms of T-Time. Table 6 shows the T-Time values of the classifier with and without HFS-MIMR.

From Table 6, we obtain the following conclusions:

Table 5
R-Time of the six feature selection methods.

Method	*F194*	*DD*	*VOC*	*Cifar100*	*SUN*	*ILSVRC65*
HFisher	**1.06**	**0.2**	**1.6**	**11.5**	**23.8**	**5.5**
HFSNM	155.9	15.04	39.36	–	–	1536
HMRMR	59.52	28.53	3214	27863	34846	11473
HiRRfam	5.35	1.71	239	8759	2220	446
HFSDK	188	28.2	700	–	–	6796
HFSLDL	11.3	1.68	45	–	–	142
LCCSHFS	7.96	3.51	1568	8727	7972	3146
HFS-MIMR	5.29	2.97	136	774	671	411

Table 6
T-Time of the classifier between with and without HFS-MIMR (s).

Dataset	The number of selected feature				
	10%	20%	30%	50%	100%
F194	77	103	123	180	366
DD	0.68	0.86	1.04	1.37	2.72
VOC	192	429	684	1206	2540
Cifar100	569	1294	2119	3869	7867
SUN	2819	5708	8716	15097	31456
ILSVRC65	421	916	1521	2651	4691

(1) HFS-MIMR selects 20% and 10% of all features from the image and protein datasets, respectively. Meanwhile, the classifier efficiency used to test these features is improved. An example is that the T-Time value is significantly reduced by an order of magnitude on the image dataset.
(2) HFS-MIMR significantly reduces the time taken for the classifier to test features. The classifier test time increases with the feature number. Therefore, HFS-MIMR can effectively improve the classifier test efficiency.

5.3. Performance comparison with different selected feature numbers

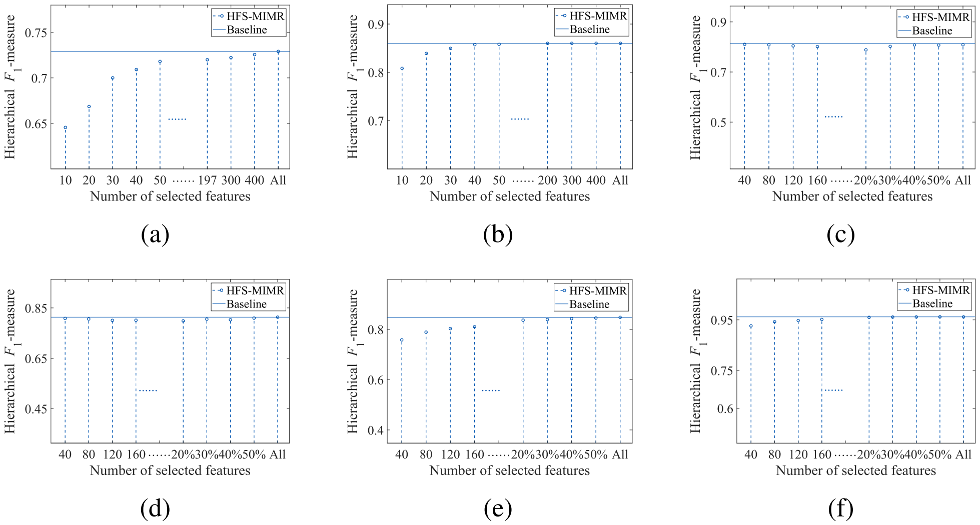
In this section, we use HFS-MIMR to select important features on different datasets and use Hier-F_1 to measure their effectiveness. The experimental design in this section is as follows: (1) All features are selected as a baseline experiment. (2) Different feature numbers are selected as sub-experiments.

Fig. 4 shows the Hier-F_1 values of HFS-MIMR for comparing the sub-experiments with the baseline experiment. From Fig. 4, we obtain the following observations:

(1) HFS-MIMR significantly reduces the feature dimension to ensure effectiveness, which is demonstrated on all experimental datasets.
(2) HFS-MIMR selects about 50 features (20%), making it equal to or even better than in the baseline experiment on *DD*. HFS-MIMR selects about 20% of all features to reach the precision of the baseline experiment on *Cifar100*, *SUN*, and *ILSVRC65*. The best performance is on *VOC*, which only selects about 50 features (1%) to achieve the baseline experiment performance.

5.4. Structural relation regularization visualization experiment

This section verifies the effectiveness of structural relation regularization. The experiments use the page-blocks dataset from the UCI machine learning repository, which has 10 features and 5 classes. The dataset is divided into training and test sets at a ratio of 4:1, as shown in Table 7.



(a) (b) (c)

(d) (e) (f)

Fig. 4. Hier-F_1 with different numbers of features. (a) *F194*, (b) *DD*, (c) *VOC*, (d) *Cifar100*, (e) *SUN*, (f) *ILSVRC65*.

Table 7
Page-blocks description.

No.	Class	Test	Training	Total
1	*text*	3,930	983	4,913
2	*horiz.Line*	263	66	329
3	*graphic*	22	6	28
4	*vert.line*	70	18	88
5	*picture*	92	23	115
Total	-	4,377	1,096	5,473

According to the semantic information of the class, we construct the page-block dataset with a tree structure as shown in Fig. 5. The following experimental results refer to 6th and 7th sibling nodes. We select features by different methods, and the correlation between the selected features is regarded as the inter-class independence measurement. In this subsection, the feature correlation is calculated by Euclidean distances.

First, we use HFS-Sparse to select features for nodes 6 and 7. Fig. 6 shows the feature weight graphs obtained by HFS-Sparse. We select important features according to the weight evaluation value. From Fig. 6a, we select the top four features {6, 5, 1, 7} for the 6th node. From Fig. 6b, we select the top four features {6, 5, 1, 10} for the 7th node.

Table 8 shows the Hier-F_1 values of different feature selection methods. We obtain the Euclidean distance between sibling nodes 6 and 7 is 2.59E + 07.

Then, we obtain the feature weight diagrams of 6th and 7th nodes using HFS-Inter, as shown in Fig. 7. We select the top-ranked features for the 6th and 7th nodes according to Fig. 7.

From Fig. 7a, we select the top four features {6, 5, 1, 7} for the 6th node. From Fig. 7b, we select the top four features {6, 5, 1, 4} for the 7th node. From Table 8, we obtain the Euclidean distance between sibling nodes 6 and 7 as 8.90E + 07, which is greater than 2.59E + 07.

To further verify the effectiveness of the selected features, we use the SVM classifier for 10-fold cross-validation. From Table 8, we obtain an HFS-Sparse Hier-F_1 value of 0.9656 and the HFS-Inter Hier-F_1 value of 0.9659. We conclude that HFS-Inter has good performance and high classification accuracy. Therefore, the structural relation regularization greatly improves the inter-class distance. Good features improve the independence between classes.

5.5. Visualization experiment of the feature relation regularization

This section proves the effectiveness of using feature relation regularization in HFS-MIMR. We use the page-blocks dataset with a tree structure as shown in Fig. 5. We use different methods to select features and then calculate the correlation between them as the redundancy criterion. The correlation between page-blocks features is calculated by the Pearson correlation coefficient as shown in Fig. 8. The following experimental results refer to the 6th node.

First, we use HFS-Sparse to select important features and obtain the feature weights for the 6th node. Fig. 9 shows the feature weight graph obtained by HFS-Sparse. From Fig. 9a, we select the top two features {6, 5}. From Fig. 9a, we obtain that the correlation coefficient between features 6 and 5 is 0.5277. Then, we use HFS-Intra to obtain the feature weights for the 6th node From Fig. 9b, we select the top two features {6, 1}. From Fig. 8, we obtain that the correlation coefficient between features 6 and 1 is 0.2676.

The SVM classifier is used to verify the performance, and Hier-F_1 values are used as the evaluation indicator. Table 9 shows the experimental performance of different feature selection methods. From Table 9, we obtain that the Hier-F_1 value of features selected by HFS-Sparse is 0.9434, while that selected by HFS-Intra is 0.9458. Therefore, HFS-Intra has a higher performance than HFS-Sparse.

We conclude that the feature relation regularization makes the selected features sparse and ensures low redundancy. Meanwhile, the feature relation regularization ensures high classification accuracy.

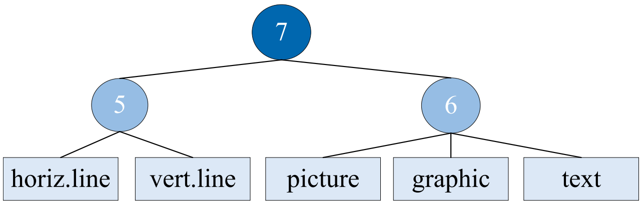


Fig. 5. Tree structure of the page-blocks dataset.

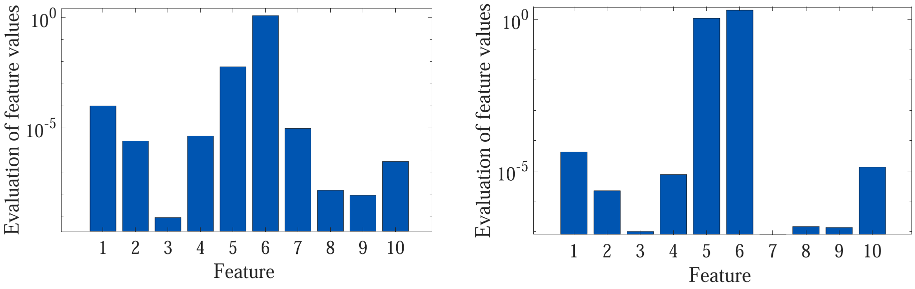


Fig. 6. Feature weight graphs obtained by HFS-Sparse for the (a) 6th node. (b) 7th nodes.

Table 8
Euclidean distances and the Hier-F_1 values of different feature selection methods.

Method	Current node	Selected features	Euclidean distance	Hier-F_1

HFS-Sparse	6th node
	7th node
		{6, 5, 1, 7}
		{6, 5, 1, 10}
			2.59E + 07	96.56

HFS-Inter	6th node
	7th node
		{6, 5, 1, 7}
		{6, 5, 1, 4}
			8.90E + 07	96.59

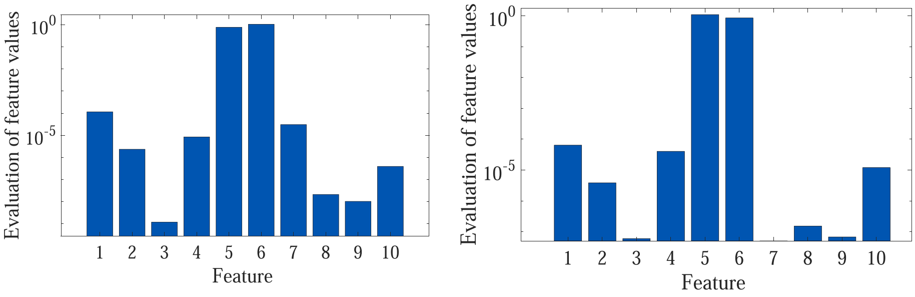
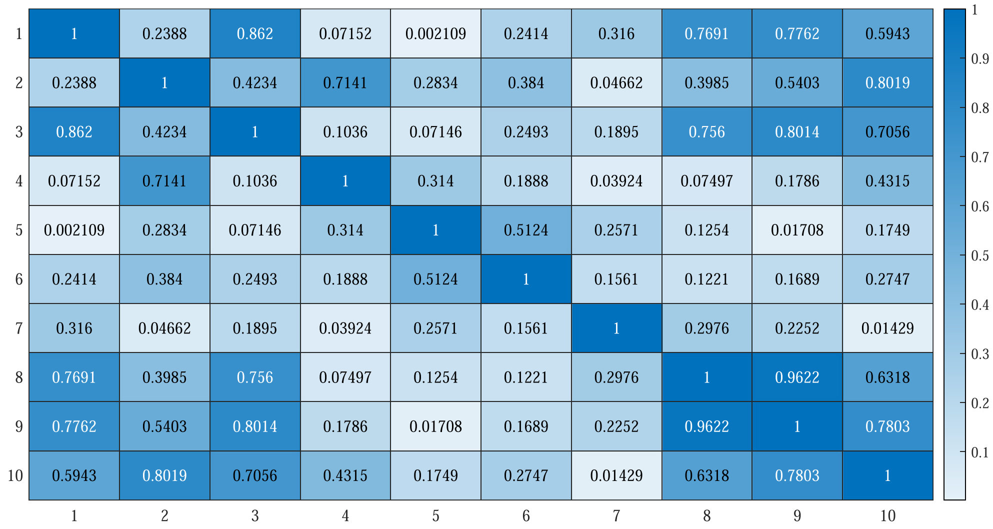


Fig. 7. Feature weight graphs of HFS-Inter for the (a) 6th node. (b) 7th nodes.



	1	2	3	4	5	6	7	8	9	10
1	1	0.2388	0.862	0.07152	0.002109	0.2414	0.316	0.7691	0.7762	0.5943
2	0.2388	1	0.4234	0.7141	0.2834	0.384	0.04662	0.3985	0.5403	0.8019
3	0.862	0.4234	1	0.1036	0.07146	0.2493	0.1895	0.756	0.8014	0.7056
4	0.07152	0.7141	0.1036	1	0.314	0.1888	0.03924	0.07497	0.1786	0.4315
5	0.002109	0.2834	0.07146	0.314	1	0.5124	0.2571	0.1254	0.01708	0.1749
6	0.2414	0.384	0.2493	0.1888	0.5124	1	0.1561	0.1221	0.1689	0.2747
7	0.316	0.04662	0.1895	0.03924	0.2571	0.1561	1	0.2976	0.2252	0.01429
8	0.7691	0.3985	0.756	0.07497	0.1254	0.1221	0.2976	1	0.9622	0.6318
9	0.7762	0.5403	0.8014	0.1786	0.01708	0.1689	0.2252	0.9622	1	0.7803
10	0.5943	0.8019	0.7056	0.4315	0.1749	0.2747	0.01429	0.6318	0.7803	1

Fig. 8. Feature correlation coefficients of the 6th node.

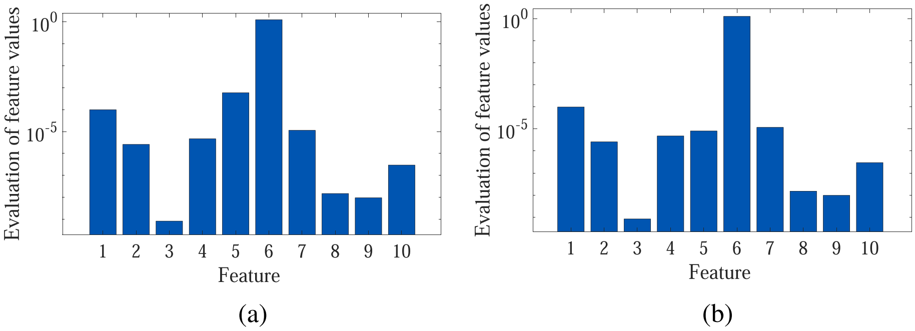


Fig. 9. The feature weight graphs of the 6^{th} node using (a) HFS-Sparse and (b) HFS-Intra.

Table 9
The Hier-F_1 values of features selected by different feature selection methods.

Method	Current node	Selected feature	Hier-F_1
HFS-Sparse	6^{th} node	{6, 5}	94.34
HFS-Intra	6^{th} node	{6, 1}	94.58

5.6. Parameter sensitivity analysis

This section analyses the influences of parameters α and β on feature selection from different datasets. The proposed method has three penalty parameters: α, β, and λ, of which λ is fixed. Parameters α and β control the punishment degree of inter-class independence and intra-class feature redundancy, respectively. We use "grid search" to adjust the parameters in the set {0.01, 0.1, 1, 10, 100, 1000}. Table 10 shows the Hier-F_1 values of features selected by HFS-MIMR with various parameter values.

From Table 10, we obtain the following conclusions:

(1) HFS-MIMR provides the best performance on most datasets when $\beta = 1$. When $\beta > 1$, the weights of redundant and irrelevant features increase, which reduces the method performance.

(2) The HFS-MIMIR method is insensitive to changes in α on most datasets. A small α value indicates that the distance between sibling classes is close, which leads to there being no obvious difference between sibling classes. It is difficult to find a compact and unique subset of local features if parameter α is too large. Both of these situations reduce the performance of the proposed HFS-MIMR method.

6. Conclusions and future work

We proposed a hierarchical feature selection method that considers structural and feature relations to maximize inter-class independence and minimize intra-class redundancy. This balances intra-class and inter-class relations in hierarchical classification tasks by structural and feature relations embedded in a unified hierarchical method. We identified differences between unrelated classes by constructing the structural relation constraint to enhance inter-class dependency. The feature relation constraint was constructed as the external knowledge to reduce intra-class redundancy. Our method effectively adjusts the balance between inter-class and intra-class relations, which results in a good performance in hierarchical classification. Future work will be extended to the graph structures and consider feature selection for cases of unbalanced data classification.

CRediT authorship contribution statement

Jie Shi: Conceptualization, Methodology, Resources, Writing – original draft. **Zhengyu Li:** Investigation, Validation, Methodology. **Hong Zhao:** Conceptualization, Methodology, Validation, Writing – review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 10
Parameter sensitivity evaluation on different datasets. (a) *Cifar100*; (b) *ILSVRC65*; (c) *SUN*; (d) *VOC*; (e) *F194*; (f) *DD*.

(a)

	α					
β	0.01	0.1	1	10	100	1000
0.01	79.14	79.14	79.14	77.06	77.09	79.10
0.1	79.13	79.14	79.17	79.17	79.18	79.13
1	79.18	79.18	79.18	79.18	79.18	79.13
10	79.14	79.14	79.09	79.12	79.01	76.85
100	79.03	78.98	79.05	79.09	79.17	79.01
1000	78.74	78.83	79.18	78.69	79.18	78.76

(b)

	α					
β	0.01	0.1	1	10	100	1000
0.01	93.26	96.01	96.01	96.00	96.01	96.09
0.1	93.26	96.00	96.01	96.00	96.01	96.07
1	95.90	96.00	96.01	95.99	96.00	96.01
10	95.99	95.99	96.00	96.00	96.01	96.01
100	95.69	95.54	95.62	95.53	96.00	96.01
1000	95.68	95.72	95.68	95.77	95.70	95.64

(c)

	α					
β	0.01	0.1	1	10	100	1000
0.01	83.54	83.53	83.54	83.50	83.55	83.50
0.1	83.54	83.58	83.57	83.54	83.35	83.52
1	83.65	83.65	83.65	83.62	83.62	83.54
10	85.33	83.58	83.47	83.43	83.54	83.58
100	82.88	82.96	82.75	82.86	82.64	82.80
1000	82.88	82.85	82.95	83.03	82.94	82.99

(d)

	α					
β	0.01	0.1	1	10	100	1000
0.01	80.36	80.36	80.33	80.35	80.43	80.11
0.1	80.25	80.25	80.23	80.24	80.36	80.26
1	80.36	80.36	80.34	80.35	80.42	80.44
10	80.20	80.20	80.16	80.25	80.25	80.27
100	79.71	79.41	79.72	79.54	79.32	79.52
1000	79.11	79.60	79.54	79.42	79.55	79.52

(e)

	α					
β	0.01	0.1	1	10	100	1000
0.01	71.71	71.71	71.53	71.55	71.55	71.67
0.1	71.71	71.71	71.48	71.57	71.57	71.60
1	71.22	71.24	70.23	70.09	70.09	70.21
10	64.74	65.23	64.15	62.62	62.62	63.29
100	69.08	68.64	68.54	68.83	68.83	68.26
1000	66.60	68.00	67.25	67.56	66.62	67.09

(f)

	α					
β	0.01	0.1	1	10	100	1000
0.01	86.12	86.12	86.18	86.18	86.18	86.07
0.1	86.12	86.12	86.18	86.18	86.18	86.07
1	86.23	86.23	86.12	86.23	86.23	86.18
10	85.02	85.02	84.96	84.63	84.63	85.57
100	85.51	86.00	85.39	86.23	86.23	82.62
1000	84.85	84.85	84.63	84.91	83.97	85.07

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 62141602 and the Natural Science Foundation of Fujian Province under Grant No. 2021J011003.

References

[1] R. Babbar, I. Partalas, E. Gaussier, M. Amini, C. Amblard, Learning taxonomy adaptation in large-scale classification, J. Mach. Learn. Res. 17 (1) (2016) 3350–3386.
[2] C. Luo, T. Li, Y. Huang, H. Fujita, Updating three-way decisions in incomplete multi-scale information systems, Inf. Sci. 476 (2019) 274–289.
[3] J. Krause, M. Stark, J. Deng, L. Feifei, 3D object representations for fine-grained categorization, in: IEEE Conference on Computer Vision Workshops, 554–561, 2013.
[4] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vision 88 (2) (2010) 303–338.
[5] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Handbook of Systemic Autoimmune Diseases 1 (4).
[6] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, A. Torralba, SUN database: Large-scale scene recognition from abbey to zoo, Int. J. Comput. Vision 119 (2016) 3–22.
[7] C.H. Ding, I. Dubchak, Multi-class protein fold recognition using support vector machines and neural networks, Bioinformatics 17 (4) (2001) 349–358.
[8] L. Wei, M. Liao, X. Gao, Q. Zou, An improved protein structural classes prediction method by incorporating both sequence and structure information, IEEE Trans. Nanobiosci. 14 (4) (2014) 339–349.
[9] R. Bai, R. Huang, Y. Qin, Y. Chen, C. Lin, HVAE: A deep generative model via hierarchical variational auto-encoder for multi-view document modeling, Inf. Sci. 623 (2023) 40–55.
[10] P. Pham, L.T. Nguyen, N.T. Nguyen, R. Kozma, B. Vo, A hierarchical fused fuzzy deep neural network with heterogeneous network embedding for recommendation, Inf. Sci. 620 (2023) 105–124.
[11] Y. Ji, S. Liu, M. Zhou, Z. Zhao, X. Guo, L. Qi, A machine learning and genetic algorithm-based method for predicting width deviation of hot-rolled strip in steel production systems, Inf. Sci. 589 (2022) 360–375.
[12] T. Li, Z. Zhan, J. Xu, Q. Yang, Y. Ma, A binary individual search strategy-based bi-objective evolutionary algorithm for high-dimensional feature selection, Inf. Sci. 610 (2022) 651–673.
[13] B. Zhao, F. Li, E. Xing, Large-scale category structure aware image categorization, Adv. Neural Inf. Process. Syst. 24 (2011) 1251–1259.
[14] S. Guo, H. Zhao, W. Yang, Hierarchical feature selection with multi-granularity clustering structure, Inf. Sci. 568 (2021) 448–462.
[15] J. Zheng, C. Luo, T. Li, H. Chen, A novel hierarchical feature selection method based on large margin nearest neighbor learning, Neurocomputing 497 (2022) 1–12.
[16] Q. Wang, J. Wan, F. Nie, B. Liu, C. Yan, X. Li, Hierarchical feature selection for random projection, IEEE Trans. Neural Networks Learn. Syst. 30 (5) (2018) 1581–1586.
[17] M. Wang, X. Hao, J. Huang, K. Wang, L. Shen, X. Xu, D. Zhang, M. Liu, Hierarchical structured sparse learning for schizophrenia identification, Neuroinformatics 18 (1) (2020) 43–57.
[18] H. Zhao, P. Zhu, P. Wang, Q. Hu, Hierarchical feature selection with recursive regularization, in: International Conference on Artificial Intelligence, 3483–3489, 2017.
[19] Q. Tuo, H. Zhao, Q. Hu, Hierarchical feature selection with subtree based graph regularization, Knowl.-Based Syst. 163 (2019) 996–1008.
[20] P. Chen, F. Li, C. Wu, Research on intrusion detection method based on Pearson correlation coefficient feature selection algorithm, J. Phys: Conf. Ser. 1757 (1) (2021) 12054.
[21] H. Wang, P. Wang, S. Deng, H. Li, Improved relief weight feature selection algorithm based on relief and mutual information, Information 12 (6) (2021) 228.
[22] C. Huang, J. Du, B. Nie, R. Yu, W. Xiong, Q. Zeng, Feature selection method based on partial least squares and analysis of traditional chinese medicine data, Computational and Mathematical Methods in Medicine 2019 (2019) 9580126.
[23] F. Nie, H. Huang, X. Cai, C. Ding, Efficient and robust feature selection via joint $\ell 2$, 1-norms minimization, Advances in Neural Information Processing Systems 23 (2010) 1813–1821.
[24] Y. Fan, B. Chen, W. Huang, J. Liu, W. Weng, W. Lan, Multi-label feature selection based on label correlations and feature redundancy, Knowl.-Based Syst. 241 (2022) 108256.
[25] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, J. Bioinf. Comput. Biol. 3 (2) (2005) 185–205.
[26] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.
[27] I. Jo, S. Lee, S. Oh, Improved measures of redundancy and relevance for mRMR feature selection, Computers 8 (2) (2019) 42–55.
[28] C. Feng, Y. Gao, J. Liu, J. Wang, D. Wang, C. Wen, Joint-norm constraint and graph-laplacian PCA method for feature extraction, BioMed Res. Int. (2017).
[29] W. Qian, Y. Xiong, J. Yang, W. Shu, Feature selection for label distribution learning via feature similarity and label correlation, Inf. Sci. 582 (2022) 38–59.
[30] C. Luo, T. Li, H. Chen, H. Fujita, Z. Yi, Incremental rough set approach for hierarchical multicriteria classification, Inf. Sci. 429 (2018) 72–87.
[31] H. Costa, L. Galvão, L.H. Merschmann, M.J. Souza, A VNS algorithm for feature selection in hierarchical classification context, Electronic Notes in Discrete Mathematics 66 (2018) 79–86.
[32] R. Cerri, R.G. Mantovani, M.P. Basgalupp, A.C. de Carvalho, Multi-label feature selection techniques for hierarchical multi-label protein function prediction, in: International Conference on Neural Networks, 1–7, 2018.
[33] H.C. Lima, F.E. Otero, L.H. Merschmann, M.J. Souza, A novel hybrid feature selection algorithm for hierarchical classification, IEEE Access 9 (2021) 127278–127292.
[34] A. Secker, M.N. Davies, A.A. Freitas, E. Clark, J. Timmis, D.R. Flower, Hierarchical classification of G-Protein-Coupled receptors with data-driven selection of attributes and classifiers, J. Data Min. Bioinf. 4 (2) (2010) 191–210.
[35] B.C. Paes, A. Plastino, A.A. Freitas, Exploring attribute selection in hierarchical classification, J. Inf. Data Manage. 26 (6) (2014) 313–317.
[36] H. Huang, H. Liu, Feature selection for hierarchical classification via joint semantic and structural information of labels, Knowl.-Based Syst. 195 (2020) 105655.
[37] X. Li, Y. Wang, R. Ruiz, A survey on sparse learning models for feature selection, IEEE Trans. Cybern. 52 (3) (2022) 1642–1660.
[38] H. Zhang, J. Wang, Z. Sun, J.M. Zurada, N.R. Pal, Feature selection for neural networks using group lasso regularization, IEEE Trans. Knowl. Data Eng. 32 (4) (2019) 659–673.
[39] P. Liu, X. Qiu, X. Huang, Adversarial multi-task learning for text classification, arXiv preprint arXiv:1704.05742.
[40] J. Han, Z. Sun, H. Hao, Selecting feature subset with sparsity and low redundancy for unsupervised learning, Knowl.-Based Syst. 86 (2015) 210–223.
[41] A. Argyriou, T. Evgeniou, M. Pontil, Multi-task feature learning, Advances in Neural Information Processing Systems 19 (2006) 41–48.
[42] P.E. Hart, D.G. Stork, R.O. Duda, Pattern classification, Wiley Hoboken, 2000.
[43] L. Grimaudo, M. Mellia, E. Baralis, Hierarchical learning for fine grained internet traffic classification, in: International Conference on Wireless Communications and Mobile Computing, 463–468, 2012.

[44] H. Zhao, Q. Hu, P. Zhu, Y. Wang, P. Wang, A recursive regularization based feature selection framework for hierarchical classification, IEEE Trans. Knowl. Data Eng. 33 (7) (2021) 2833–2846.

[45] X. Liu, Y. Zhou, H. Zhao, Robust hierarchical feature selection driven by data and knowledge, Inf. Sci. 551 (2021) 341–357.

[46] Y. Lin, H. Liu, H. Zhao, Q. Hu, X. Zhu, X. Wu, Hierarchical feature selection based on label distribution learning, IEEE Trans. Knowl. Data Eng. 33 (7) (2022) 2667.

[47] Y. Lin, S. Bai, H. Zhao, S. Li, Q. Hu, Label-correlation-based common and specific feature selection for hierarchical classification, J. Software 33 (7) (2022) 2667–2682.

[48] O. Dekel, J. Keshet, Y. Singer, Large margin hierarchical classification, in: International Conference on Machine Learning, 27–34, 2004.

[49] A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, I. Androutsopoulos, Evaluation measures for hierarchical classification: A unified view and novel approaches, Data Min. Knowl. Disc. 29 (3) (2015) 820–865.

[50] A.V. Aho, J.E. Hopcroft, J.D. Ullman, On finding lowest common ancestors in trees, SIAM J. Comput. 5 (1) (1976) 115–132.