Expert Systems With Applications 255 (2024) 124588

Contents lists available at ScienceDirect

Expert Systems With Applications

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

DMTFS-FO: Dynamic multi-task feature selection based on flexible loss and orthogonal constraint

Yang Zhang, Jie Shi, Hong Zhao [*]

School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, 363000, China
The Key Laboratory of Data Science and Intelligence Application, Minnan Normal University, Zhangzhou, Fujian, 363000, China

ARTICLE INFO

Keywords:
Multi-task feature selection
Dynamic task
Hierarchical classification
Orthogonal constraint

ABSTRACT

Multi-task feature selection (MTFS) has been proven effective for reducing the curse of dimensionality in large-scale classification. Many existing MTFS methods assume that all tasks learn concurrently in a static environment without considering the dynamism of tasks in real-world scenarios. However, new tasks emerge dynamically in practical applications, meaning that the aforementioned static assumption is insufficient. In this paper, we construct a dynamic multi-task feature selection framework to achieve feature reduction for constantly arriving new tasks. First, we modify the traditional mapping by shifting from hard labels to soft labels. Unlike the conventional rigid mapping, the new flexible loss function changes the direct mapping strategy to an indirect one. Second, we use the orthogonal regularization term to constrain the independent relationship between new and old tasks. This ensures that the selected relevant features for new tasks differ from prior tasks. Finally, we integrate the flexible loss and the orthogonal regularization term in the dynamic multi-task feature selection framework. Our method outperforms nine other advanced feature selection methods in terms of effectiveness and efficiency across six datasets. For example, the ACC value of our method is almost 1% higher than the next-best method on the large-scale *SUN* dataset.

1. Introduction

Multi-task feature selection (MTFS) is a powerful technique for solving high-dimensional feature problems in massive-class classification tasks, which operates by ensuring that multiple subtasks learn simultaneously (Guo, Zhao, & Yang, 2021). From the perspective of features, datasets in the big data era contain a large number of redundant features, which affect classification performance (Chen, Fu, Yao, Guo, Plant, & Wang, 2023; Chen, Yang, Li, Wang, & Qian, 2022). Furthermore, the growing number of classes increases the computation time and storage cost for classification. MTFS can overcome these issues and is widely applied in many practical fields, such as medical diagnosis (Shao, Peng, Zu, Wang, & Zhang, 2020), bioinformatics (Lan & Vucetic, 2011), speech emotion recognition (Kalhor & Bakhtiari, 2021), and image processing (Zhao, Peng, & He, 2020).

Traditional feature selection methods mostly overcome high-dimensional feature problems by analyzing relationships in datasets, which can be catalogized into feature-relationship-based and label-relationship-based methods. The former utilizes the relationships between features to reduce irrelevant and redundant features. They can be divided into similarity-based methods (He, Lin, Lin, & Wang, 2024; Jia, Deng, Wang, & Wang, 2024; Samareh Jahani, Saberi Movahed, Eftekhari, Aghamollaei, & Tiwari, 2024) and information-theory-based methods (Gong, Li, Zhang, Zhang, & Wang, 2024; Lim & Kim, 2021; Liu, Hu, & Zhu, 2023), according to their measurement technique. Label-relationship-based feature selection analyzes the correlation of labels to find potential rules and mine label information. Some methods improve performance by exploring the pairwise label correlations (Gao, Li, & Hu, 2023; Li, Hu, & Gao, 2022). In addition, the method based on pseudo-labels explores more complex correlations among all labels, beyond label pairs to mine the most informative features (Fan et al., 2022; Fan, Chen, et al., 2024; Liu, Chen, Li & Li, 2023). These methods have reduced feature dimensionality by utilizing the relationships between features and labels in the dataset. Nevertheless, they ignore the effects of the hierarchical structure relationships between classes, which makes it difficult to solve dimensional disaster problems in large-scale classification tasks (Hu et al., 2018).

MTFS exploits the hierarchical structure relationship of labels to solve the issue of high-dimensional features in multi-task environments. This large-scale complex classification task can be divided into smaller sub-tasks according to the hierarchical structure. Further, the hierarchical structure mimics the human thinking pattern—from abstract concepts to concrete instances—arranging classes in tree-like structures

[*] 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.eswa.2024.124588
Received 19 January 2024; Received in revised form 19 June 2024; Accepted 22 June 2024
Available online 3 July 2024

or directed acyclic graphs. MTFS can use the hierarchical structure to select its discriminative feature subset for each task, thus simplifying the classification difficulty. Zhao, Hu, Zhu, Wang, and Wang (2021) used inter-task relations to provide useful assistance for the feature selection of large-scale classes, selecting strong discriminative features for each task. Subsequently, many scholars have leveraged the dual relationship constraints between tasks to optimize the multi-task feature selection process on this basis (Huang & Liu, 2020; Shi & Zhao, 2023). Recently, some methods have been proposed to employ single-constraint as auxiliary information, including similarity constraints and independence constraints. For example, Liu, Lin, Wang, Guo, and Chen (2023) proposed an MTFS method based on label distributions constrained by the consistency of parent–child distributions. Moreover, for independent constraints, Shi, Li, and Zhao (2023) provided a feature selection method that maximizes the difference in feature subsets between tasks.

The methods mentioned above utilize dual or single-dependency constraints between tasks to analyze the relationships between tasks to simplify the classification difficulty. However, they learn all tasks simultaneously in a hypothetical static environment, which is inconsistent with actual dynamic conditions (Belouadah, Popescu, & Kanellos, 2021; Van de Ven, Tuytelaars, & Tolias, 2022).

In this paper, we propose a dynamic framework for multi-task feature selection that can adapt to new task learning, which conforms to the dynamic nature of real-world tasks. First, we reconstruct the mapping of features from hard labels into soft labels within flexible loss, which contains more potential information. Second, we restrict the dependency among new and old tasks using an orthogonal constraint, making the feature subset more pertinent and discriminative to different tasks. Under the orthogonal constraint, the new task selects its specific feature with the help of the knowledge learned from the old tasks. This further enables tasks to obtain feature subsets corresponding to themselves that differ from other tasks. Finally, we incorporate flexible loss in tasks and relationship constraints among new and old tasks, embedding them into the dynamic multi-task feature selection framework. Further, we leverage the sparse $l_{2,1}$-norm to select the most informative and representative features, reducing redundant information in the features.

The effectiveness of dynamic multi-task feature selection based on flexible loss and orthogonal constraint (DMTFS-FO) is verified through comparison with other methods modified from advanced feature selection algorithms. The experiments are conducted on protein and image datasets. We use various indicators, including the ACC, Hier-F_1, F_{LCA}, and TIE metrics, to measure the effectiveness. The experimental results indicate that the DMTFS-FO method can effectively select feature subsets for new tasks and improve performance through flexible loss and independent constraints.

The main contributions of this paper are as follows:

- We propose a basic dynamic multi-task feature selection framework to select relevant discriminative features for new tasks, which effectively realizes the dimensional reduction of dynamic data in real-world scenarios.
- We present flexible loss to alter the traditional direct mapping to indirect mapping. Unlike traditional direct mapping, we incorporate soft labels to provide potential knowledge, which is beneficial for selecting relevant features.
- We investigate the relationships between tasks and introduce orthogonal regularization. It helps to select the feature subsets of new tasks more accurately by maximizing the independence between new and old tasks.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 proposes the DMTFS-FO framework with the corresponding optimization method. Section 4 introduces the experimental setup and discusses the experimental results of the DMTFS-FO algorithm. Finally, Section 5 summarizes this paper and provides future research directions.

2. Related work

We first review traditional feature selection, followed by multi-task feature selection (MTFS).

2.1. Traditional feature selection

Traditional feature selection methods have received extensive attention for reducing the computational and storage overhead caused by irrelevant and redundant features (Fan, Liu, et al., 2024). These methods include feature-relationship-based and label-relationship-based methods. Feature-relationship-based methods focus on utilizing feature relationships to eliminate irrelevant and redundant features. He et al. (2024) introduced a feature similarity constraint using the Gaussian kernel similarity matrix. Jia et al. (2024) leveraged the similarity between features to potentially identify and exclude redundant features. Similarly, Samareh Jahani et al. (2024) reduced feature redundancy by considering the local correlation of features through orthogonalization. In addition, some methods use statistical metrics to measure the relationship between features. For example, Lim and Kim (2021) considered the pairwise dependency of features to reduce feature redundancy. Furthermore, Gong et al. (2024) combined the Pearson correlation coefficient and normalized mutual information to characterize the dependency between features. Through sparse learning, Liu, Hu, and Zhu (2023) selected the most informative feature while considering feature relations through mutual information.

Label-relationship-based methods investigate the label relationship as equally crucial in feature selection. Some methods have investigated the relationships between pairs of labels. Li, Hu, and Gao (2022) adopted label-level regularization to consider pairwise label correlations. Gao et al. (2023) proposed a method that shares and preserves potential label structures while considering the relationships between label pairs. Furthermore, many methods mine the label relationship information through pseudo-label matrices to explore the relationships among all labels. For instance, Fan et al. (2022) preserved the structure of the original label space through a low-dimensional pseudo-label matrix and explored label relations. Furthermore, Liu, Chen, et al. (2023) constructed latent semantic labels using the similarity matrix between labels. Fan, Chen, et al. (2024) combined the logical label matrix and non-negative label relaxation matrix to construct a pseudo label matrix, learning the correlation among labels. Based on these methods, we adopt a flexible label matrix to incorporate more knowledge. The aforementioned methods remove irrelevant and redundant features in high-dimensional features by exploring different relationships.

However, these methods assume that classes are independent of each other, which is unsuitable for direct application to large-scale classification tasks. The proposed method utilizes label hierarchical relations for investigating the relationships among classes to reduce the complexity of classification tasks.

2.2. Multi-task feature selection

Recently, MTFS has received widespread research attention due to its advantages in simplifying the difficulty of large-scale classification tasks (Liu, Sheng, et al., 2020). Some MTFS methods divide complex tasks into simple subtasks and explore their relationships, driven by label hierarchical tree structure information (Liu, Lin, et al., 2023; Wang et al., 2017). MTFS methods have different constraints on the relationships between tasks, including methods based on dual-dependency and single-dependency constraints. The former focuses more on utilizing inter-task relationships to select specific features of subtasks. For instance, Zhao et al. (2021) explored the tree structure information of the label space to recursively select the feature set of each task by constraining each task through sibling and parent–child dependency. Based on this method, Huang and Liu (2020) combined the similarity score between labels as semantic regularization based

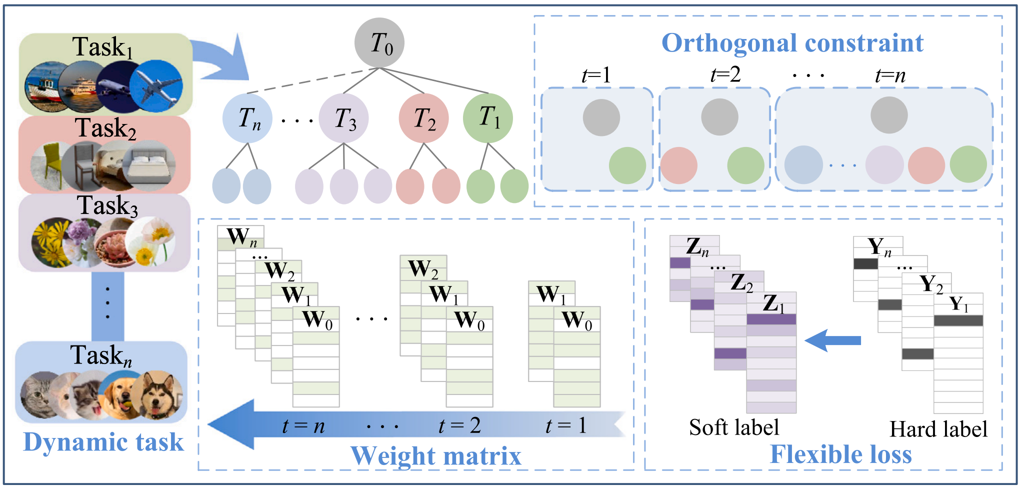


Fig. 1. Framework of the DMTFS-FO method.

on considering the dependency of tasks. Furthermore, Lin, Liu, et al. (2022) proposed an MTFS method based on label distribution learning that takes advantage of the degree of dependency of parent–child and sibling relationships. Subsequently, Shi and Zhao (2023) presented a feature selection based on structural manifold learning and hierarchical information to automatically balance the dependency between tasks.

The MTFS methods based on single-dependency constraints exploit task dependencies as auxiliary information, including similarity constraints and independent constraints. From the perspective of similarity constraints, Tuo, Zhao, and Hu (2019) extended the parent–child similarity relationship between tasks to a graph structure to implement bidirectional dependency constraints for classes. Liu, Lin, et al. (2023) introduced a method that replaces the original logical labels with label distributions constrained by the consistency of parent–child distributions. From the perspective of independence constraints, Liu, Lin, et al. (2020) ensured the diversity of feature subsets between sibling nodes by punishing the similarity between the tasks. Furthermore, Shi et al. (2023) improved the independence between unrelated tasks by investigating the class relationships at the same layer in the hierarchy while minimizing intra-task feature redundancy.

Although these MTFS algorithms can obtain relatively compact and strong discriminative features under closed static assumptions, this assumption is unreasonable due to the dynamicity and complexity of tasks in the real world (Belouadah et al., 2021; Van de Ven et al., 2022). In comparison to the previous MTFS methods, the proposed method constrains the relationships between new and old tasks through orthogonal representation to adapt to the new tasks.

3. Dynamic multi-task feature selection framework

We present a dynamic multi-task feature selection method based on flexible loss and orthogonal constraints (DMTFS-FO) in detail and introduce an optimization process for DMTFS-FO.

3.1. Overview of DMTFS-FO

The framework of DMTFS-FO is designed as shown in Fig. 1 and consists mainly of the following three parts.

(1) The basic dynamic multi-task feature selection framework (DMTFS) selects relevant features for new tasks.

(2) The flexible loss term adjusts and improves the traditional feature mapping.

(3) The orthogonal regularization restricts the relations of new and old tasks and aids in selecting task-specific features.

3.2. A basic DMTFS framework

DMTFS selects features for new tasks in a dynamic environment. Nowadays, with the proliferation of classes, tasks become increasingly complex and massive. We leverage the hierarchical structure of labels to decompose complex global tasks into relatively simple subtasks to explore the relations between tasks (Liu, Zhou, & Zhao, 2021). First, we explain the idea of the proposed method using Example 1.

Example 1. Fig. 2 shows the process of dynamic multi-task feature selection proposed by our method. Initially, the samples in the dataset are classified into their corresponding hierarchical tree-structured tasks, which are different and independent from each other. Then, the proposed method trains a “Vehicle” task to select features useful for classifying cars and ships. Subsequently, a second task is introduced and relevant features are selected for classifying furniture images. Finally, the “Animal” task enters the training, and our method selects discriminative features to distinguish cats, pandas, and dogs. The proposed method continuously learns new tasks while minimizing the impact on the knowledge of old tasks, achieving knowledge accumulation. This adaptive learning method is significant in real-world scenarios due to data is constantly changing and updating.

Then, we define the symbols used in the DMTFS framework. We set Q as the number of new tasks and t as the number of iterations. In each iteration t, the new task will enter the training process according to its position in the tree structure. At the tth iteration, let the sum of the new and old tasks number to N_t, where $N_t = N_{t-1} + Q$. Let $\mathbf{X}_i^{(t)} = [\mathbf{x}_i^1; \cdots; \mathbf{x}_i^j; \cdots; \mathbf{x}_i^{n_i}] \in \mathbf{R}^{n_i \times d}$ be a feature matrix of ith task with n_i instances lie in a d-dimensional feature space. For the feature matrix of the ith task, we represent the corresponding label matrix as $\mathbf{Y}_i^{(t)} = [\mathbf{y}_i^1; \cdots; \mathbf{y}_i^j; \cdots; \mathbf{y}_i^{n_i}] \in \{0,1\}^{n_i \times m}$, which m indicates the maximum number of classes for all tasks. The feature weight matrix of the ith task learnt from feature matrix $\mathbf{X}_i^{(t)}$ and label matrix $\mathbf{Y}_i^{(t)}$ is present as $\mathbf{W}_i^{(t)} = [\mathbf{w}_i^1; \cdots; \mathbf{w}_i^j; \cdots; \mathbf{w}_i^d] \in \mathbf{R}^{d \times m}$, which $\mathbf{w}_i^j$ represents the importance of the jth feature. The $\mathbf{W}_i^{(t)}$ matrix is derived after several iterations of optimizing the objective equation. Sorting the matrix $\|\mathbf{W}_i^{(t)}\|_2$ in descending order, with the higher value indicating the corresponding feature is more critical for classification. The original data is replaced by selecting and retaining the top K features, thereby achieving dimensionality reduction while preserving classification accuracy.

Sparse learning is an effective method with great performance and interpretability in feature selection (Li, Wang, & Ruiz, 2022). Therefore, we propose a DMTFS framework based on sparse learning, aiming at

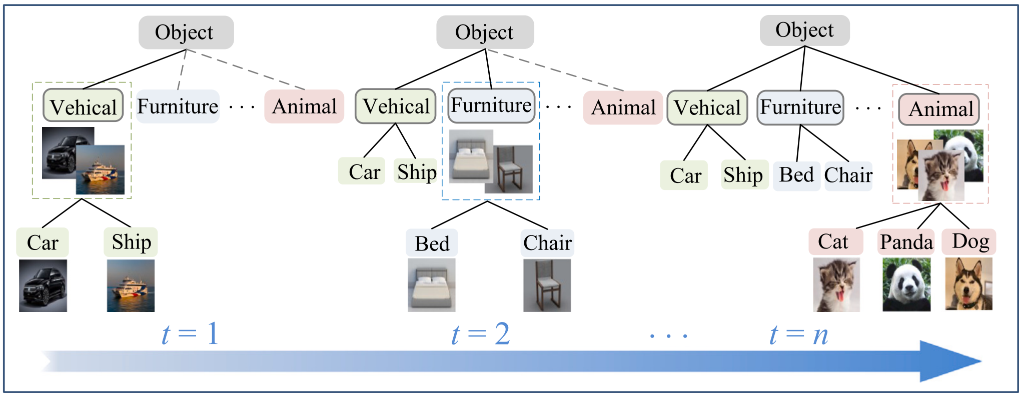


Fig. 2. An example of dynamic training for different tasks.

jointly reducing irrelevant features and minimizing fitting errors in each task. The formula is expressed as follows:

$$\min_{\mathbf{W}_0^{(t)},\ldots,\mathbf{W}_{N_t}^{(t)}} \sum_{t=1}^{T}\sum_{i=0}^{N_t} \|\mathbf{X}_i^{(t)}\mathbf{W}_i^{(t)} - \mathbf{Y}_i^{(t)}\|_F^2 + \underbrace{\beta\|\mathbf{W}_i^{(t)}\|_{2,1}}_{inner-task\ sparsity}, \tag{1}$$

where the symbol $\|\cdot\|_F^2$ is the least squares loss term for fitting linear error between predicted labels $\mathbf{X}_i^{(t)}\mathbf{W}_i^{(t)}$ and real labels $\mathbf{Y}_i^{(t)}$. The symbol $\|\cdot\|_{2,1}$ is the $l_{2,1}$-norm regularization term, which is the common and effective sparse learning norm in multi-task feature selection for it has convexity and structured sparsity. Parameter β is for adjusting the sparsity. The high sparsity of $\mathbf{W}_i^{(t)}$ represents that it has more zero elements, thereby reducing the weight of unimportant features.

3.3. A DMTFS based on flexible loss

Labels are the training target of the supervised feature selection algorithm, which can help the algorithm judge the contribution of each feature to model prediction (Qian, Xiong, Yang, & Shu, 2022). Traditional feature selection methods usually employ the least squares regression loss to map features to hard labels that are not either 0 or 1. Nevertheless, this mapping is absolute, and we modify the hard labels in the original mapping to soft labels and construct the indirect mapping as a flexible loss. Inspired by Zhang, Luo, Li, Zhou, and Li (2019), we introduce the matrix $\mathbf{Z}_i^{(t)}$ as a soft label matrix of the ith task during iteration t, which converts hard label $\mathbf{Y}_i^{(t)}$ to the floating-point number. We buffer the original direct mapping by utilizing soft labels. Therefore, we use the Frobenius norm to define the flexible loss function to establish the indirect mapping of the feature matrix to the label matrix. The flexible loss $L(\mathbf{W}_i^{(t)},\mathbf{Z}_i^{(t)})$ is mathematically represented as follows:

$$L(\mathbf{W}_i^{(t)},\mathbf{Z}_i^{(t)}) = \sum_{t=1}^{T}\sum_{i=0}^{N_t} \|\mathbf{X}_i^{(t)}\mathbf{W}_i^{(t)} - \mathbf{Z}_i^{(t)}\|_F^2 + \alpha\|\mathbf{Y}_i^{(t)} - \mathbf{Z}_i^{(t)}\|_F^2, \tag{2}$$

where α is used to adjust the buffering degree of mapping.

The flexible loss has the following advantages: (1) Soft labels can incorporate more potential knowledge in the iterative learning process and more flexibility compared with hard labels. (2) Flexible loss changes the traditional feature direct mapping to indirect, which can capture more complex relations from features to labels and improve performance.

We reconstruct the loss of the basic DMTFS method into flexible loss, and the new objective function is as follows:

$$\min_{\mathbf{W}_0^{(t)},\ldots,\mathbf{W}_{N_t}^{(t)}} \sum_{t=1}^{T}\sum_{i=0}^{N_t} \underbrace{\underbrace{\|\mathbf{X}_i^{(t)}\mathbf{W}_i^{(t)} - \mathbf{Z}_i^{(t)}\|_F^2}_{least\ squares\ loss} + \underbrace{\alpha\|\mathbf{Y}_i^{(t)} - \mathbf{Z}_i^{(t)}\|_F^2}_{soft\ label\ regularizer}}_{flexible\ loss} + \underbrace{\beta\|\mathbf{W}_i^{(t)}\|_{2,1}}_{inner-task\ sparsity}. \tag{3}$$

3.4. A DMTFS based on orthogonal constraint

Dynamic multi-task learning aims to enhance the effectiveness of learning new tasks by leveraging the experience gained from previously learned old tasks (Guo, Haque, Huang, Yeung, & FeiFei, 2018). We assume tasks that have entered the optimization process are old tasks. Gradually, new tasks that are different from the old ones increasingly appear in the learning process. We observe that the features between new and old tasks on the same layer are dissimilar or even completely different to distinguish their specific subclasses. We use the following example to illustrate this situation.

Example 2. Continuing with Example 1, we intuitively explain the independence of features among different tasks in Fig. 3. The features “wheel size” and “capacity” are suitable for classifying samples from the old task “Vehicle”, but they fail to serve as distinguishing features for the next new task “Furniture”. Similarly, the features of the task “Furniture”, such as the “material” and “style”, are not applicable to the new task “Animal”. Besides, the features of the old tasks “Vehicle” and “Furniture” are both useless for the new task “Animal”. Overall, the features of new and old tasks are different and independent.

Each new task is distinct from the previous old tasks and unrelated. Therefore, we employ independence constraints on the feature weights of new and old tasks. In many independent constraints, the orthogonal constraint regularization term is simple and effective, which can maximize the independence of features between new and old tasks (Zhao & Yu, 2019). We impose the orthogonal constraint regularization term $R(\mathbf{W}_i^{(t)},\mathbf{W}_j^{(t)})$ on feature weights of new and old tasks, which is defined as:

$$R(\mathbf{W}_i^{(t)},\mathbf{W}_j^{(t)}) = \sum_{t=1}^{T}\sum_{i=1}^{N_t}\sum_{j\in E_i^{(t)}} \|\mathbf{W}_j^{(t)\mathrm{T}}\mathbf{W}_i^{(t)}\|_F^2, \tag{4}$$

where j represents the old task independent of the ith new task, and $E_i^{(t)}$ is the set of existing old tasks independent of the ith new task during the tth iteration. Matrices $\mathbf{W}_i^{(t)}$ and $\mathbf{W}_j^{(t)}$ represent the feature weight matrix of the new and old tasks.

The orthogonal constraint for new and old tasks has the following benefits: (1) The orthogonal constraint term can ensure the features of new and old tasks are irrelevant. (2) New tasks can more accurately select their specific feature subsets under the constraints.

Based on all the above discussion, we combine flexible loss and the orthogonal constraint term. The final objective function of DMTFS-FO

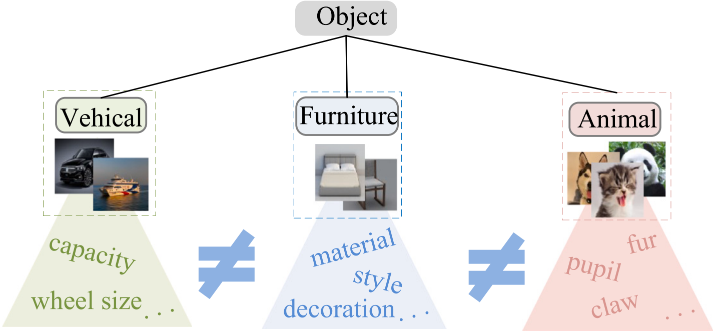


Fig. 3. An example of feature independence among different tasks.

can be formulated as:

$$\min_{\mathbf{W}_0^{(t)},\ldots,\mathbf{W}_{N_t}^{(t)}} \sum_{t=1}^{T} \left(\underbrace{\sum_{i=0}^{N_t} \Big(\underbrace{\|\mathbf{X}_i^{(t)}\mathbf{W}_i^{(t)} - \mathbf{Z}_i^{(t)}\|_F^2}_{least\ squares\ loss} + \underbrace{\alpha\|\mathbf{Y}_i^{(t)} - \mathbf{Z}_i^{(t)}\|_F^2}_{soft\ label\ regularizer}}_{\text{flexible loss}} + \underbrace{\beta\|\mathbf{W}_i^{(t)}\|_{2,1}}_{\text{inner-task sparsity}} \Big) \right. \\ \left. + \underbrace{\gamma \sum_{i=1}^{N_t} \sum_{j\in E_i} \|\mathbf{W}_j^{(t)^{\mathrm{T}}}\mathbf{W}_i^{(t)}\|_F^2}_{\text{inter-task independence}} \right), \tag{5}$$

where γ balances the degree of feature independence among new and old tasks. As the iteration t increases, the number of tasks N_t gradually increases until all tasks enter the learning process. New and old tasks select task-specific features under orthogonal constraints.

3.5. Method analysis and discussion

We have a comprehensive analysis of the two different dynamic task scenarios of the model. In the first case, when the number of dynamic new tasks in the proposed method is equal to the number of all intermediate nodes N in the tree structure, then all tasks learn simultaneously. The method degenerates into a static multi-task feature selection, represented by the following formula:

$$\min_{\mathbf{W}_0,\ldots,\mathbf{W}_N} \sum_{i=0}^{N} (\|\mathbf{X}_i\mathbf{W}_i - \mathbf{Z}_i\|_F^2 + \alpha\|\mathbf{Y}_i - \mathbf{Z}_i\|_F^2 + \beta\|\mathbf{W}_i\|_{2,1}) \\ + \gamma \sum_{i=1}^{N} \sum_{j\in E_i} \|\mathbf{W}_j^{\mathrm{T}}\mathbf{W}_i\|_F^2, \tag{6}$$

where E_i represents tasks at the same hierarchy level.

In the second case, the method has only one static root node task to distinguish all classes when the number of dynamic tasks in the proposed method is zero. The method degenerates from multi-task feature selection to traditional planar feature selection, which is expressed by the following formula:

$$\min_{\mathbf{W}} \|\mathbf{X}\mathbf{W} - \mathbf{Z}\|_F^2 + \alpha\|\mathbf{Y} - \mathbf{Z}\|_F^2 + \beta\|\mathbf{W}\|_{2,1}. \tag{7}$$

Compared with the two degradation methods mentioned above, DMTFS-FO has the following advantages: (1) Unlike the static MTFS method, DMTFS-FO simulates the constantly emerging dynamic tasks in reality and utilizes the knowledge of old tasks to dynamically influence the selection of new task features. (2) Unlike traditional flat feature selection, DMTFS-FO does not directly classify all classes. The proposed method fully utilizes the spatial structure information of labels to break down large-scale tasks into related small-scale dynamic tasks. It recursively selects each more targeted and compact feature subset.

3.6. Optimization algorithm

We provide a detailed introduction to the optimization process of the objective function in the state at the tth iteration. For the objective function Eq. (5), we need to optimize the variables $\mathbf{Z}_i^{(t)}$ and $\mathbf{W}_i^{(t)}$ separately.

We optimize $\mathbf{Z}_i^{(t)}$ first. By taking the derivative of Eq. (5) with respect to $\mathbf{Z}_i^{(t)}$ to zero, we have following result for $\mathbf{Z}_i^{(t)}$ as

$$\mathbf{Z}_i^{(t)} = \frac{1}{\alpha+1}(\mathbf{X}_i^{(t)}\mathbf{W}_i^{(t)} + \alpha\mathbf{Y}_i^{(t)}). \tag{8}$$

Then, we optimize $\mathbf{W}_i^{(t)}$. We divide the objective equation into two parts: dynamic internal node tasks and static root node tasks. Therefore, we reformulate the final objective function Eq. (5) as

$$J(\mathbf{W}_0^{(t)}, \mathbf{W}_1^{(t)}, \ldots, \mathbf{W}_{N_t}^{(t)}) = \|\mathbf{X}_0^{(t)}\mathbf{W}_0^{(t)} - \mathbf{Z}_0^{(t)}\|_F^2 + \alpha\|\mathbf{Y}_0^{(t)} - \mathbf{Z}_0^{(t)}\|_F^2 + \beta\|\mathbf{W}_0^{(t)}\|_{2,1} \\ + \sum_{i=1}^{N_t} (\|\mathbf{X}_i^{(t)}\mathbf{W}_i^{(t)} - \mathbf{Z}_i^{(t)}\|_F^2 \\ + \alpha\|\mathbf{Y}_i^{(t)} - \mathbf{Z}_i^{(t)}\|_F^2 + \beta\|\mathbf{W}_i^{(t)}\|_{2,1} \\ + \gamma \sum_{j\in E_i} \|\mathbf{W}_j^{(t)^{\mathrm{T}}}\mathbf{W}_i^{(t)}\|_F^2). \tag{9}$$

In the first part, the static root node formula is represented as

$$J(\mathbf{W}_0^{(t)}) = \|\mathbf{X}_0^{(t)}\mathbf{W}_0^{(t)} - \mathbf{Z}_0^{(t)}\|_F^2 + \alpha\|\mathbf{Y}_0^{(t)} - \mathbf{Z}_0^{(t)}\|_F^2 + \beta\|\mathbf{W}_0^{(t)}\|_{2,1}, \tag{10}$$

where the direct derivation of $\|\cdot\|_{2,1}$ is challenging due to its non-smoothness. We refer to the method in Argyriou, Evgeniou, and Pontil (2006) to solve this problem. The derivative formula for the $l_{2,1}$-norm-norm is as follows:

$$\frac{\partial(\|\mathbf{W}_i^{(t)}\|_{2,1})}{\partial\mathbf{W}_i^{(t)}} = \frac{\partial(Tr(\mathbf{W}_i^{(t)^{\mathrm{T}}}\mathbf{D}_i^{(t)}\mathbf{W}_i^{(t)}))}{\partial\mathbf{W}_i^{(t)}} = 2\mathbf{D}_i^{(t)}\mathbf{W}_i^{(t)}, \tag{11}$$

where $\mathbf{D}_i^{(t)} \in \mathbf{R}^{d\times d}$ is a diagonal matrix. Its jth diagonal element value is defined as

$$D^{i\ (t)}_{jj} = \frac{1}{2\|\mathbf{w}^{i\ (t)}_j\|_2}, \tag{12}$$

where we set $D^{i\ (t)}_{jj} = \epsilon$ when $\mathbf{w}^{i\ (t)}_j = 0$, and parameter ϵ is a very small constant as the threshold. We make the derivative of Eq. (10) with respect to $\mathbf{W}_0^{(t)}$ to zero as

$$\frac{\partial J(\mathbf{W}_0^{(t)})}{\partial\mathbf{W}_0^{(t)}} = (\mathbf{X}_0^{(t)^{\mathrm{T}}}\mathbf{X}_0^{(t)} + \beta\mathbf{D}_0^{(t)})\mathbf{W}_0^{(t)} - \mathbf{X}_0^{(t)^{\mathrm{T}}}\mathbf{Z}_0^{(t)} = 0. \tag{13}$$

Therefore, we obtain $\mathbf{W}_0^{(t)}$ as

$$\mathbf{W}_0^{(t)} = (\mathbf{X}_0^{(t)^{\mathrm{T}}}\mathbf{X}_0^{(t)} + \beta\mathbf{D}_0^{(t)})^{-1}(\mathbf{X}_0^{(t)^{\mathrm{T}}}\mathbf{Z}_0^{(t)}). \tag{14}$$

In the second part, the optimization formula for dynamic tasks is expressed as

$$J(\mathbf{W}_1^{(t)},\ldots,\mathbf{W}_{N_t}^{(t)})=\sum_{i=1}^{N_t}(\|\mathbf{X}_i^{(t)}\mathbf{W}_i^{(t)}-\mathbf{Z}_i^{(t)}\|_F^2+\alpha\|\mathbf{Y}_i^{(t)}-\mathbf{Z}_i^{(t)}\|_F^2 \\ +\beta\|\mathbf{W}_i^{(t)}\|_{2,1}+\gamma\sum_{j\in E_i^{(t)}}\|\mathbf{W}_j^{(t)^{\mathrm{T}}}\mathbf{W}_i^{(t)}\|_F^2). \tag{15}$$

We take the derivative of Eq. (15) with respect to $\mathbf{W}_i^{(t)}$ to obtain:

$$\frac{\partial J(\mathbf{W}_1^{(t)},\ldots,\mathbf{W}_{N_t}^{(t)})}{\partial \mathbf{W}_i^{(t)}}=\mathbf{X}_i^{(t)^{\mathrm{T}}}(\mathbf{X}_i^{(t)}\mathbf{W}_i^{(t)}-\mathbf{Z}_i^{(t)})+\beta\mathbf{D}_i^{(t)}\mathbf{W}_i^{(t)} \\ +\gamma\sum_{j\in E_i^{(t)}}(\mathbf{W}_j^{(t)}\mathbf{W}_j^{(t)^{\mathrm{T}}})\mathbf{W}_i^{(t)}. \tag{16}$$

By setting the value of Eq. (16) to zero, we have:

$$\mathbf{W}_i^{(t)}=(\mathbf{X}_i^{(t)^{\mathrm{T}}}\mathbf{X}_i^{(t)}+\beta\mathbf{D}_i^{(t)}+\gamma\sum_{j\in E_i^{(t)}}\mathbf{W}_j^{(t)}\mathbf{W}_j^{(t)^{\mathrm{T}}})^{-1}(\mathbf{X}_i^{(t)^{\mathrm{T}}}\mathbf{Z}_i^{(t)}). \tag{17}$$

We present the iterative process of the method in detail in Algorithm 1.

Algorithm 1 The algorithm of DMTFS-FO.

```
Input:
(1) Data matrices $\mathbf{X}_i^{(t)} \in \mathbf{R}^{n_i\times d}$ and $\mathbf{Y}_i^{(t)} \in \mathbf{R}^{n_i\times m}$ of tasks of the $i$-th task at $t$-iteration;
(2) Parameters: $\alpha$, $\beta$, and $\gamma$;
(3) The maximal iteration number $T$;
(4) Number of learning tasks $N_t \in \{1,2,\cdots,N\}$, where $N$ is the total number of tasks.
Output: Weight matrix $\mathbf{W}^{(T)} \in \mathbf{R}^{d\times m}$.
1:  Set iteration number $t = 1$;
2:  Initialize $\mathbf{W} \in \mathbf{R}^{n\times d}$ randomly;
3:  while $t \le T$ do
4:      for $i = 0 : N_t$ do
5:          Compute the flexible label matrix $\mathbf{Z}_i^{(t)}$ according to Eq. (8);
6:          Compute the diagonal matrix $\mathbf{D}_i^{(t)}$ according to Eq. (12);
7:      end for
8:      Update $\mathbf{W}_0^{(t)}$: $\mathbf{W}_0^{(t+1)} = (\mathbf{X}_0^{(t)^{\mathrm{T}}}\mathbf{X}_0^{(t)} + \beta\mathbf{D}_0^{(t)})^{-1}(\mathbf{X}_0^{(t)^{\mathrm{T}}}\mathbf{Z}_0^{(t)})$;
9:      for $i = 1 : N_t$ do
10:         Update $\mathbf{W}_i^{(t)}$: $\mathbf{W}_i^{(t+1)} = (\mathbf{X}_i^{(t)^{\mathrm{T}}}\mathbf{X}_i^{(t)} + \beta\mathbf{D}_i^{(t)} + \gamma\sum_{j\in E_i^{(t)}}\mathbf{W}_j^{(t)}\mathbf{W}_j^{(t)^{\mathrm{T}}})^{-1}(\mathbf{X}_i^{(t)^{\mathrm{T}}}\mathbf{Z}_i^{(t)})$;
11:     end for
12:     Update $\mathbf{W}^{(t+1)} = \left\{\mathbf{W}_0^{(t+1)}, \mathbf{W}_1^{(t+1)}, \cdots, \mathbf{W}_{N_t}^{(t+1)}\right\}$;
13:     $t = t + 1$;
14: end while
15: return $\mathbf{W}^{(T)}$;
```

The iterative update of the feature weight $\mathbf{W}_i^{(t)}$ of dynamic tasks in line 10 largely determines the time complexity of the algorithm. The complexity associated with each iteration for the feature weights of a task is articulated in terms of $O(dn_im + d^2m + d^2n_i)$, where n_i is the number of rows for the ith node, m represents the number of classes for the task, and d represents the number of features. Since $\mathbf{X}^T\mathbf{X}$ only needs to be calculated once, the time complexity across all tasks is $O(d^2n)$. Let T represent the total number of iterations, then the final time complexity of DMTFS-FO is $O(T(dnm + d^2\ m) + d^2n)$.

4. Experimental setup and results analysis

We first introduce the experimental setup, including the experimental setting, comparison methods, and dataset descriptions. Then, we present and discuss the experimental results to verify the effectiveness of the proposed method, including: (1) Effectiveness comparison with other methods; (2) Influence of different terms; (3) Influence of dynamic task order; (4) Convergence analysis; and (5) Parameter sensitivity analysis.

4.1. Experimental setting

We describe the experimental settings, which are detailed below:

(1) **The optimal parameters** are obtained by grid search. We discover that the method performs best when $\beta = 10$, so the value of β is fixed to ten. The tuning range by the grid-search strategy for parameters α and γ is set to [0.1, 1, 10, 100, 1000]. By randomly combining values within the given range and evaluating the experimental results of each combination separately, we identify the optimal parameters as follows. For protein datasets, the optimal parameters are $\alpha = 100$ and $\gamma = 1$. For image datasets, the optimal parameters are $\alpha = 100$ and $\gamma = 10$.

(2) **The evaluation indicators** to measure the performance of the methods include ACC, Hier-F_1, F_{LCA}, and TIE. ACC represents the proportion of correctly predicted samples to all samples. Hier-F_1 (Cai & Hofmann, 2007) assesses using ancestors and descendants of real and predicted classes. F_{LCA} (Schieber & Vishkin, 1988) measures the distance between predicted and real labels by the closest common ancestor. TIE (Dekel, Keshet, & Singer, 2004) quantifies errors based on edges between real and predicted labels in the hierarchy.

(3) **Implementation details** of the experiments are as follows:

We maintain the same settings as in Ref. (Zhao et al., 2021), selecting the top 10% and 20% features of the protein and image datasets. For the fairness of the experiment, a top-down classifier and linear kernel were used to test the performance of the selected feature subset, and all experimental results were averaged under a 10-fold cross-validation strategy.

4.2. Comparison methods

We modify nine advanced MTFS methods into dynamic MTFS as comparative methods. The detailed explanation of the comparison method is as follows:

- **DFS-MIMR** is a dynamic MTFS method modified from HFS-MIMR (Shi et al., 2023), which utilizes task independence and minimizes the redundancy of intra-task features, dynamically selecting new task features. We follow the optimization parameters given in the paper of HFSCF to set the parameters $\alpha = 0.1$, $\beta = 0.1$, $\gamma = 10$ on the protein datasets, and $\alpha = 1$, $\beta = 1$, $\gamma = 100$ on the image datasets.
- **DFSRR** is a dynamic MTFS method modified from HFSRR (Zhao et al., 2021), which leverages the parent–child and sibling relations between tasks to gradually select relevant features for new tasks. We follow the optimization parameters given in the paper of HFSRR to set the parameters $\alpha = 0.1$, $\beta = 0.1$, $\gamma = 10$ on the protein datasets, and $\alpha = 1$, $\beta = 1$, $\gamma = 10$ on the image datasets.
- **DFSDK** is a dynamic MTFS method modified from HFSDK (Liu et al., 2021), which divides multiple tasks through hierarchical data labels, using upper-bound hinge loss constraints to eliminate data outliers. We follow the optimization parameters given in the paper of HFSDK to set the parameters $\alpha = 0.1$, and $\beta = 0.1$ on all datasets.
- **DFSLDL** is a dynamic MTFS method modified from HFSLDL (Lin, Liu, et al., 2022), which utilizes label distribution and class correlation to alleviate class imbalance. We follow the optimization parameters given in the paper of HFSLDL to set the parameters $\alpha = 0.1$, $\beta = 0.1$, $\lambda_1 = 10$, $\lambda_2 = 0.1$, and $\lambda_3 = 0.1$ on all datasets.
- **DFSCF** is a dynamic MTFS method modified from HFSCF (Liu, Lin, et al., 2020), which considers intra-class consistency and differences between tasks to select representative features. We follow the optimization parameters given in the paper of HFSCF to set the parameters $\alpha = 0.005$, $\beta = 1$, and $\gamma = 10$ on all datasets.

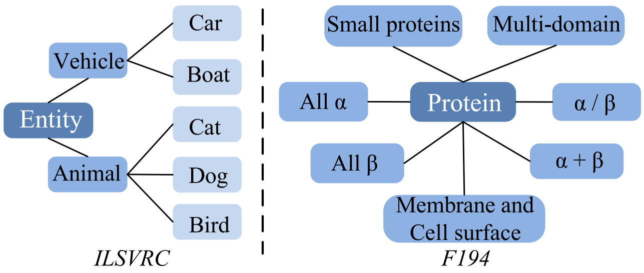


Fig. 4. The class hierarchy of the *ILSVRC* and *F194* datasets.

- **DFSLE** is a dynamic MTFS method modified from HFSLE (Liu, Lin, et al., 2023), which compensates for semantic gaps in hierarchical structures through label enhancement. We follow the optimization parameters given in the paper of HFSLE to set the parameters $\alpha = 0.1$, $\beta = 0.1$, $\gamma = 10$, and $\theta = 0.1$ on all datasets.
- **DFSCN** is a dynamic MTFS method modified from HFSCN (Liu & Zhao, 2021), which reduces the inter-layer error propagation problem by using the capped l_2-norm based loss. We follow the optimization parameters given in the paper of HFSCN to set the parameters $\alpha = 10$, $\gamma = 10$, and $epsi = 0.1$ on all datasets.
- **DFSGKFS** is a dynamic method modified from HFSGKFS (Qiu & Zhao, 2022), which uses the fuzzy rough set method based on Hausdorff distance to select features for new and old tasks. We follow the optimization parameter given in the paper of HFSGKFS to set the parameter $\delta = 0.9$ on all datasets.
- **LCCSDFS** is a dynamic method modified from LCCSHFS (Lin, Bai, Zhao, Li, & Hu, 2022), which leverages recursive regularization to select corresponding intrinsic and common features for each task in the hierarchical structure. We follow the optimization parameter given in the paper of LCCSHFS to set the parameter $\alpha = 0.001$, $\lambda = 100$ on all datasets.

All experiment results are re-implemented on a computer with an Intel(R) Xeon(R) Gold 6126, 2.60 GHz CPU, and the 64-bit Windows 10 operating system. The code can be obtained from the following link: https://github.com/fhqxa/DMTFS-FO.

4.3. Experimental dataset descriptions

We leverage six datasets in the experiments, including protein and image datasets. The protein datasets comprise *DD* (Ding & Dubchak, 2001) and *F194* (Wei, Liao, Gao, & Zou, 2014). *DD* comprises 27 real classes and 3,625 samples, including 3,020 training samples and 605 test samples. *F194* contains 8,525 samples and 473 features, with 194 real classes. There are four image datasets, including *VOC* (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010), *SUN* (Xiao, Hays, Ehinger, Oliva, & Torralba, 2010), *AWA* (Lampert, Nickisch, & Harmeling, 2009), and *ILSVRC* (Krause, Stark, Deng, & Li, 2013). *VOC* is a standard image recognition and object detection benchmark. It comprises 4,096 features and 20 classes, with 3,437 training and 3,539 test samples. *SUN* comprises 324 real classes and 4,096 features, consisting of 45,109 training samples and 22,556 test samples. *AWA* consists of animal attribute datasets, including 6,405 training and 3,202 testing samples. Its feature dimension is 252, and the class number is ten. *ILSVRC* has 57 classes and 24,191 samples, including 12,346 training samples and 11,845 testing samples. These datasets we used have distinct hierarchical tree structures. For instance, Fig. 4 presents the class hierarchy of the protein dataset *F194* and the image dataset *ILSVRC*. The detailed information of the experimental datasets are listed in Table 1. The “Task” column is the number of all tasks in the datasets.

Table 1
Descriptions of hierarchical datasets.

Dataset	Type	Feature	Training	Test	Class	Task
DD	Protein	473	3020	605	27	5
F194	Protein	473	7105	1420	194	8
VOC	Image	4096	3437	3539	20	4
SUN	Image	4096	45109	22,556	324	19
AWA	Image	252	6405	3202	10	7
ILSVRC	Image	4096	12,346	11,845	57	8

Table 2
The ACC results of different methods on experimental datasets (↑).

Method	*DD*	*F194*	*VOC*	*SUN*	*AWA*	*ILSVRC*	Avg.rank
DFS-MIMR	68.77	34.65	58.72	61.99	24.24	85.07	4.08
DFSRR	68.77	34.44	58.75	62.86	24.02	85.05	4.00
DFSDK	65.96	31.20	58.35	62.19	24.14	82.16	7.67
DFSLDL	68.61	33.87	58.46	62.91	21.02	85.09	5.17
DFSCF	68.77	34.44	58.49	62.96	24.17	84.99	3.83
DFSLE	68.77	33.31	58.38	62.12	23.68	85.11	5.42
DFSCN	68.27	33.66	58.21	62.51	23.49	82.26	7.00
DFSGKFS	67.79	20.35	57.25	60.66	20.86	84.45	9.17
LCCSDFS	68.11	32.89	–	62.06	–	**85.60**	7.50
DMTFS-FO	**68.94**	**34.79**	**58.89**	**62.97**	**24.61**	85.17	**1.00**

Table 3
The Hier-F_1 results of different methods on experimental datasets (↑).

Method	*DD*	*F194*	*VOC*	*SUN*	*AWA*	*ILSVRC*	Avg.rank
DFS-MIMR	85.96	**71.71**	79.81	83.90	57.32	95.91	3.25
DFSRR	85.96	71.64	79.80	84.09	57.17	95.90	4.00
DFSDK	84.63	69.74	79.66	83.66	57.14	95.17	7.75
DFSLDL	86.07	71.34	79.70	84.15	56.26	95.92	4.08
DFSCF	85.96	71.57	79.66	84.12	57.32	95.87	4.33
DFSLE	85.95	71.20	79.62	83.72	57.05	95.92	6.00
DFSCN	85.95	71.10	79.53	83.88	56.96	95.10	7.42
DFSGKFS	85.46	63.17	79.25	82.87	56.26	95.65	9.08
LCCSDFS	85.68	70.54	–	83.63	–	**96.04**	7.67
DMTFS-FO	**86.17**	71.69	**79.85**	**84.15**	**57.39**	95.95	**1.42**

Table 4
The F_{LCA} results of different methods on experimental datasets (↑).

Method	*DD*	*F194*	*VOC*	*SUN*	*AWA*	*ILSVRC*	Avg.rank
DFS-MIMR	82.57	64.07	76.42	77.01	49.46	92.32	3.58
DFSRR	82.57	63.97	76.42	77.42	49.29	92.30	4.08
DFSDK	80.97	61.94	76.23	76.94	49.31	90.86	7.50
DFSLDL	82.57	63.63	76.29	77.48	47.68	92.33	4.50
DFSCF	82.57	63.93	76.27	77.48	49.44	92.26	4.25
DFSLE	82.57	63.37	76.21	76.93	49.09	92.33	5.92
DFSCN	82.40	63.44	76.12	77.19	48.97	90.84	7.17
DFSGKFS	81.99	55.04	75.66	75.94	47.63	91.95	9.17
LCCSDFS	82.21	62.90	–	76.88	–	**92.58**	7.67
DMTFS-FO	**82.73**	**64.11**	**76.49**	**77.50**	**49.63**	92.37	**1.17**

4.4. Effectiveness comparison with other methods

We present and discuss the effectiveness of DMTFS-FO and other comparison methods under different evaluation metrics. First, we evaluate the performance of the methods in terms of the ACC, Hier-F_1, and F_{LCA} indicators. The experimental results of the nine methods are shown in Tables 2–4. The “↑” indicates “the smaller, the better”, and “-” indicates that the method cannot be trained on the dataset. Black bold text represents the best result. It can observe the following conclusions from Tables 2–4.

(1) The effectiveness of DMTFS-FO outperforms most comparative methods that depend on inter-task relationship constraints, including DFS-MIMR, DFSRR, DFSLDL, and DFSCF. We deduce that the flexible loss can better adjust the difference between predicted and real labels and assist in selecting relevant features.

(2) The performance of DMTFS-FO is better than that of many other methods. For example, the ACC values of our method exceed DFSDK by

Table 5
TIE of different DMTFS methods on experimental datasets (↓).

Method	*DD*	*F194*	*VOC*	*SUN*	*AWA*	*ILSVRC*	Avg.rank
DFS-MIMR	0.8427	**1.6972**	1.1616	1.2878	3.4147	0.3272	3.25
DFSRR	0.8427	1.7014	1.1636	1.2727	3.4266	0.3279	4.17
DFSDK	0.9221	1.8155	1.1700	1.3069	3.4234	0.3862	7.50
DFSLDL	0.8361	1.7197	1.1673	1.2680	3.4990	0.3262	4.08
DFSCF	0.8427	1.7056	1.1706	1.2704	3.4147	0.3306	4.42
DFSLE	0.8428	1.7282	1.1732	1.3025	3.4359	0.3267	6.00
DFSCN	0.8430	1.7338	1.1771	1.2892	3.4434	0.3919	7.50
DFSGKFS	0.8723	2.2099	1.1935	1.3703	3.4990	0.3477	9.08
LCCSDFS	0.8592	1.7676	–	1.3096	–	**0.3168**	7.67
DMTFS-FO	**0.8295**	1.6986	**1.1599**	**1.2676**	**3.4084**	**0.3240**	**1.33**

Table 6
Running time (s) of different methods on experimental datasets.

Method	*DD*	*F194*	*VOC*	*SUN*	*AWA*	*ILSVRC*
DFS-MIMR	1.54	3.48	72.46	455.84	0.45	179.93
DFSRR	3.67	8.76	254.67	1240.56	0.72	468.17
DFSDK	9.47	45.81	122.20	15336.63	31.53	1640.92
DFSLDL	1.45	2.46	115.58	578.52	0.78	235.43
DFSFC	15.32	40.63	177.15	3964.23	33.73	631.90
DFSLE	20.78	117.32	211.07	11515.49	164.38	1393.84
DFSCN	3.42	11.76	60.38	1585.88	9.87	262.72
LCCSDFS	5.49	8.59	–	7520.77	–	3670.10
DFSGKF	2.08	80.77	2929.05	86168.94	11.19	24980.56
DMTFS-FO	**0.82**	**2.95**	**50.99**	**332.26**	**0.28**	**21.40**

2.98% on *DD* and 3.01% on *ILSVRC*. The Hier-F_1 value exceeds the that of DFSGKFS method by 8.52% on *F194*. These results demonstrate that our method exhibits significantly superior performance compared with existing methods.

(3) On the *F194* dataset, the Hier-F_1 result of DMTFS-FO is slightly inferior to the best method by 0.02%, ranking second. This is due to the uneven distribution of nodes within the hierarchical structure of the *F194* dataset, which is affected by common ancestors.

Then, we compared the results of different methods using TIE evaluation metrics. To account for the impact of the sample size, we normalize the TIE value to enable better comparison across datasets. Table 5 demonstrates the normalized TIE values of different methods on datasets, where "↓" indicates "the smaller, the better".

We have the following observations from Table 5.

(1) The TIE values of the *AWA* dataset are larger than those of others. We infer that *AWA* has a sparser tree structure than other datasets, posing a relatively large distance between the predicted and real labels when misclassified. DMTFS-FO outperforms other comparative algorithms on *AWA*, indicating its advantages in processing sparse tree-structured datasets.

(2) DMTFS-FO performs well on types of protein and image datasets. DMTFS-FO is slightly behind DFS-MIMR on the *F194* dataset but still ranks first on average across all datasets and is far superior to DFSGKFS.

Finally, we perform statistical analysis on the experimental results of all evaluation indicators to further demonstrate the effectiveness of DMTFS-FO.

In purpose to determine the existence of performance differences, we adopt Friedman's test (Friedman, 1937) on the experimental results of different indicators. In the experiment, there are k algorithms and N datasets (k=10, N=6). We calculate the average rank R_i of each method in the ith dataset. Under the null assumption of rank equivalence of all methods, the Friedman statistic is expressed as $F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}$, where $\chi_F^2 = \frac{12N}{k(k+1)}(\sum_{i=1}^{k} R_i^2 - \frac{k(k+1)^2}{4})$. We obtain F_F values under different evaluation indicators, including $F_F = 7.818$ for ACC, $F_F = 9.410$ for Hier-F_1, $F_F = 8.574$ for F_{LCA}, and $F_F = 9.063$ for TIE. These values are greater than the critical value $F(k-1,(k-1)(N-1)) = F(9-1,(9-1)(6-1)) = 2.096$ at the significance level of $\alpha = 0.05$, therefore the null hypothesis is false.

We further utilize the Bonferroni Dunn test (Dunn, 1961) to compare the degree of performance differences between different methods. The critical distance is presented as $CD_\alpha = q_\alpha\sqrt{\frac{k(k+1)}{6N}}$. When the significant level $\alpha = 0.05$, there is $q_\alpha = 2.6380$. We calculate that $CD = 4.611$ and show the statistical results of the Bonferroni Dunn test with different evaluation indicators in Fig. 5. The results indicate that the DMTFS-FO method is statistically better than most other methods under different evaluation indicators and ranks first.

4.5. Efficiency comparison

In this section, we use running time to compare the efficiency of nine feature selection methods and DMTFS-FO. The running time of each method on six datasets is shown in Table 6. It can be seen in Table 6 that our method is superior to other methods on all datasets, especially for the *SUN* dataset with more tasks. For instance, the running time of our method is significantly shorter than that of the DFSGKF method across all datasets. These results indicate that our method exhibits good performance and superior efficiency.

4.6. Influence of different terms

We validate and analyze the impacts of the flexible loss term and orthogonal constraint regularizer in DMTFS-FO. The combinations of different components in DMTFS-FO include:

(1) DMTFS-F denotes the method with flexible loss and without the orthogonal constraint regularizer;

(2) DMTFS-O denotes the method with the orthogonal constraint and without flexible loss terms.

Fig. 6 shows the experimental results of DMTFS-F, DMTFS-O, and DMTFS-FO on six datasets in terms of the Hier-F_1. The following insights can be obtained from this figure.

(1) DMTFS-FO compares favorably with DMTFS-F in terms of *F194*. The performance of DMTFS-FO on *F194* is slightly lower than that of DMTFS-F, as the distribution of tree structure nodes in *F194* is uneven and affected by ancestor nodes. Furthermore, DMTFS-FO is superior to DMTFS-O on all datasets. For example, DMTFS-FO is about 0.22% and 0.17% better than *DD* and *VOC*, respectively.

(2) The overall performance of DMTFS-F exceeds that of DMTFS-O, indicating that the contribution of DMTFS-F to DMTFS-FO exceeds that of DMTFS-O guided by the orthogonal constraint. This further demonstrates the importance of flexible indirect mapping and the potential information contained. However, the performance of DMTFS-O is superior to that of DMTFS-F in some datasets with significant task differences, such as *DD* and *SUN*. This indicates that the orthogonal constraint term among tasks is effective.

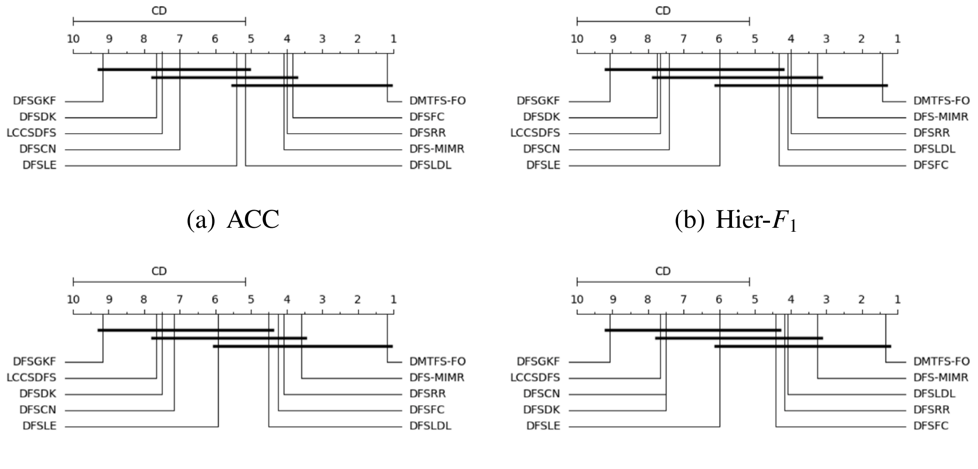
(3) DMTFS-FO performs best on most datasets, indicating that the combination of flexible loss and orthogonal constraint can effectively select the optimal feature subset. In summary, the experimental results are better when we combine the flexible loss with orthogonal constraint among tasks rather than only considering one of them.

4.7. Influence of dynamic task order

We validate the influence of the learning order of the task on DMTFS-FO. We compare three types of task learning sequences: positive, inverse, and random.

Table 7 shows the Hier-F_1 results on varying the order of tasks.

This table shows that the value of Hier-F_1 only fluctuates slightly among the three task orders on the datasets, and there is no regularity between them. This indicates that the task order has little impact on our dynamic multi-task feature selection algorithm.



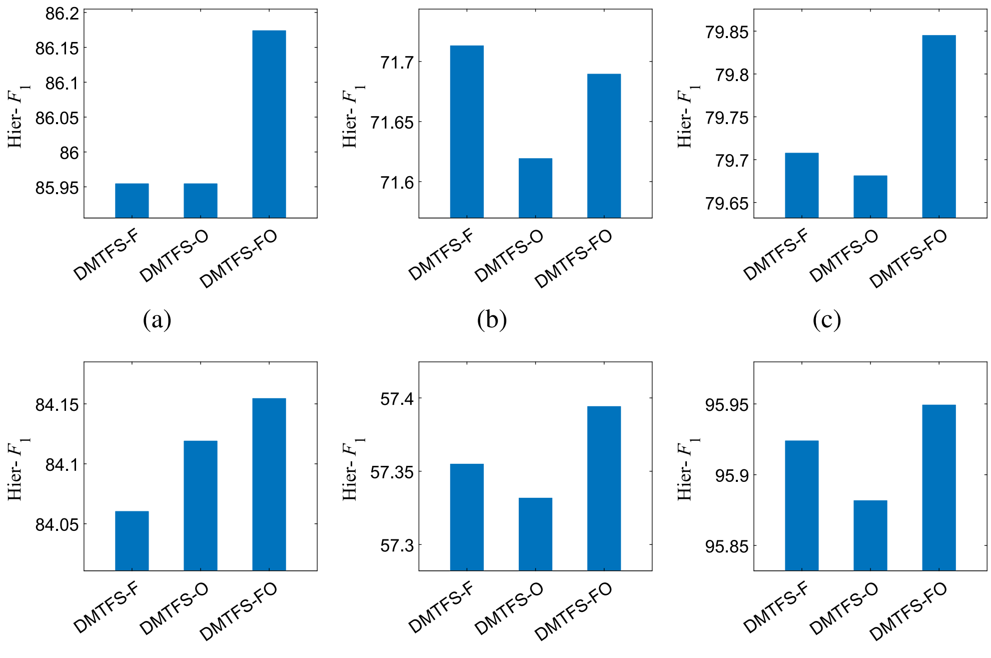
(a) ACC

(b) Hier-F_1

(c) F_{LCA}

(d) TIE

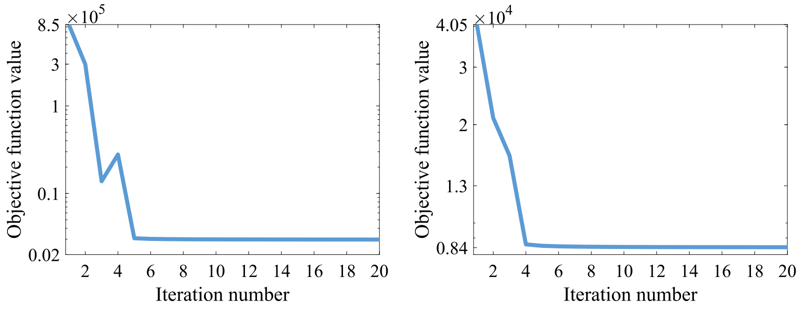
Fig. 5. Performance comparing the DMTFS-FO with other methods using the Bonferroni-Dunn test.



(a) (b) (c)

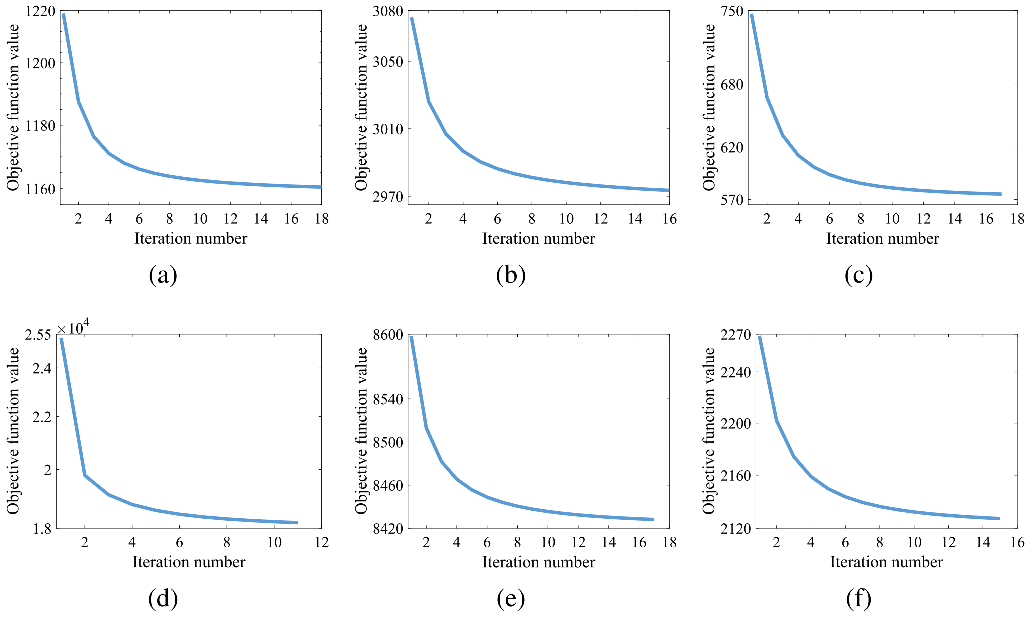
(d) (e) (f)

Fig. 6. Ablation experiments on different datasets in terms of Hier-F_1. (a) *DD*; (b) *F194*; (c) *VOC*; (d) *SUN*; (e) *AWA*; (f) *ILSVRC*.



(a) (b)

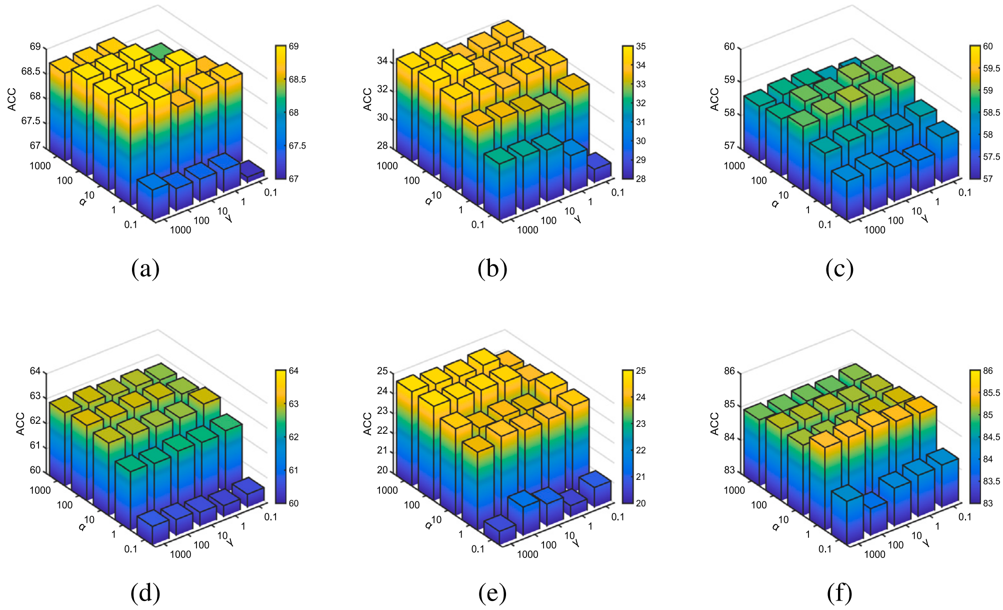
Fig. 7. Convergence curves of DMTFS-FO on (a) *F194* and (b) *AWA*.



(a) (b) (c)

(d) (e) (f)

Fig. 8. Convergence curves of DMTFS-FO after the participation of all new tasks in learning. (a) *DD*; (b) *F194*; (c) *VOC*; (d) *SUN*; (e) *AWA*; (f) *ILSVRC*.



(a) (b) (c)

(d) (e) (f)

Fig. 9. Parameter sensitivity analysis of DMTFS-FO on experimental datasets.

Table 7
Hier-F_1 of three type tasks learning orders on different datasets (↑).

Method	*DD*	*F194*	*VOC*	*SUN*	*AWA*	*ILSVRC*
Positive	86.17	71.69	79.85	84.15	57.39	95.95
Inverse	85.96	71.60	79.75	84.19	57.31	95.95
Random	85.96	71.71	79.70	84.11	57.32	95.91

4.8. Convergence analysis

We investigate the convergence of the DMTFS-FO on different datasets. The maximal iteration number is set as T=20. Fig. 7 shows the convergence curve based on the objective function value in Eq. (5) on the *F194* and *AWA* datasets. As can be seen in Fig. 7, we obtain the observation that the value of the objective function significantly drops with each iteration, gradually converging to a stable value

We ignore the dynamic addition process of new tasks to observe the convergence situation better. Fig. 8 shows the convergence of the objective function values after all new tasks participate in learning. From this figure, it can be seen that the DMTFS-FO algorithm converges with different objective function values on each dataset and gradually stabilizes to reach the convergence value with an increasing number of iterations.

4.9. Parameter sensitivity analysis

We analyze the influences of varying parameters α and γ on different datasets. The DMTFS-FO method has three penalty parameters: α, β, and γ. Parameters α and γ control the buffering degree in the mapping and adjust the independence among tasks, respectively. Parameter β operates the sparsity of features. We fix β = 10 and alternately vary one of α and γ in the set {0.1, 1, 10, 100, 1000}. Fig. 9 shows the ACC values of features selected by DMTFS-FO with various parameter value combinations.

From Fig. 9, the following observations can be obtained:

(1) DMTFS-FO performs optimally on most datasets when $\alpha = 100$. A large α renders soft labels ineffective in buffering the mapping. A small α value makes the mapping bias too large, which reduces performance.

(2) DMTFS-FO is not sensitive to parameter γ. A smaller γ value indicates that tasks are closely related, making it difficult to accurately select specific features for each task. In contrast, selecting a relevant and compact feature subset for each task is difficult when the γ value is large.

5. Conclusions and future work

In this paper, we propose a dynamic multi-task feature selection framework based on a combination of the flexible loss and orthogonal constraint (DMTFS-FO) that adapts to selecting features for new tasks. We introduce soft labels that are more potentially informative and flexible, which changes the direct mapping of features to labels to an indirect mapping technique. Orthogonal regularizers constrain the feature weights of the old and new tasks to maximize their differences, which helps new tasks select task-specific features. This guarantees that the feature subsets of each task are pertinent and discriminative to itself. DMTFS-FO solves the feature selection problem for dynamic new tasks while utilizing the flexible loss and the orthogonal constraint among tasks, which can be effectively applied in real-world dynamic scenarios.

However, the proposed method is limited in exploring the relationships between dynamic tasks. Future work will constrain complex relationships between dynamic tasks through manifold learning and regularization. The proposed method can also be better refined based on the design of adaptive parameters to adjust the relationships between tasks.

CRediT authorship contribution statement

Yang Zhang: Conceptualization, Methodology, Software, Writing – original draft. **Jie Shi:** Validation, Methodology, Writing. **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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 62376114 and the Natural Science Foundation of Fujian Province, China under Grant No. 2021J011003.

References

Argyriou, A., Evgeniou, T., & Pontil, M. (2006). Multi-task feature learning. *Advances in Neural Information Processing Systems*, *19*, 41–48.

Belouadah, E., Popescu, A., & Kanellos, I. (2021). A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, *135*, 38–54.

Cai, L., & Hofmann, T. (2007). Exploiting known taxonomies in learning overlapping concepts. In *International joint conference on artificial intelligence*: *vol. 7*, (pp. 708–713).

Chen, Z., Fu, L., Yao, J., Guo, W., Plant, C., & Wang, S. (2023). Learnable graph convolutional network and feature fusion for multi-view learning. *Information Fusion*, *95*, 109–119.

Chen, Y., Yang, X., Li, J., Wang, P., & Qian, Y. (2022). Fusing attribute reduction accelerators. *Information Sciences*, *587*, 354–370.

Dekel, O., Keshet, J., & Singer, Y. (2004). Large margin hierarchical classification. In *International conference on machine learning* (pp. 27–34).

Ding, C. H., & Dubchak, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, *17*(4), 349–358.

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association*, *56*(293), 52–64.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, *88*, 303–338.

Fan, Y., Chen, B., Huang, W., Liu, J., Weng, W., & Lan, W. (2022). Multi-label feature selection based on label correlations and feature redundancy. *Knowledge-Based Systems*, *241*, Article 108256.

Fan, Y., Chen, X., Luo, S., Liu, P., Liu, J., Chen, B., et al. (2024). Label relaxation and shared information for multi-label feature selection. *Information Sciences*, *671*, Article 120662.

Fan, Y., Liu, J., Tang, J., Liu, P., Lin, Y., & Du, Y. (2024). Learning correlation information for multi-label feature selection. *Pattern Recognition*, *145*, Article 109899.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, *32*(200), 675–701.

Gao, W., Li, Y., & Hu, L. (2023). Multilabel feature selection with constrained latent structure shared term. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(3), 1253–1262.

Gong, H., Li, Y., Zhang, J., Zhang, B., & Wang, X. (2024). A new filter feature selection algorithm for classification task by ensembling pearson correlation coefficient and mutual information. *Engineering Applications of Artificial Intelligence*, *131*, Article 107865.

Guo, M., Haque, A., Huang, D., Yeung, S., & FeiFei, L. (2018). Dynamic task prioritization for multitask learning. In *European conference on computer vision* (pp. 270–287).

Guo, S., Zhao, H., & Yang, W. (2021). Hierarchical feature selection with multi-granularity clustering structure. *Information Sciences*, *568*, 448–462.

He, Z., Lin, Y., Lin, Z., & Wang, C. (2024). Multi-label feature selection via similarity constraints with non-negative matrix factorization. *Knowledge-Based Systems*, Article 111948.

Hu, Q., Wang, Y., Zhou, Y., Zhao, H., Qian, Y., & Liang, J. (2018). Review on hierarchical learning methods for large-scale classification task. *Scientia Sinica*, *48*(5), 487–500.

Huang, H., & Liu, H. (2020). Feature selection for hierarchical classification via joint semantic and structural information of labels. *Knowledge-Based Systems*, *195*, Article 105655.

Jia, Q., Deng, T., Wang, Y., & Wang, C. (2024). Discriminative label correlation based robust structure learning for multi-label feature selection. *Pattern Recognition*, Article 110583.

Kalhor, E., & Bakhtiari, B. (2021). Multi-task feature selection for speech emotion recognition: Common speaker-independent features among emotions. *Journal of AI and Data Mining*, *9*(3), 269–282.

Krause, J., Stark, M., Deng, J., & Li, F. (2013). 3D object representations for fine-grained categorization. In *IEEE international conference on computer vision workshops* (pp. 554–561).

Lampert, C. H., Nickisch, H., & Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *IEEE conference on computer vision and pattern recognition* (pp. 951–958).

Lan, L., & Vucetic, S. (2011). Improving accuracy of microarray classification by a simple multi-task feature selection filter. *International Journal of Data Mining and Bioinformatics*, *5*(2), 189–208.

Li, Y., Hu, L., & Gao, W. (2022). Label correlations variation for robust multi-label feature selection. *Information Sciences*, *609*, 1075–1097.

Li, X., Wang, Y., & Ruiz, R. (2022). A survey on sparse learning models for feature selection. *IEEE Transactions on Cybernetics*, *52*(3), 1642–1660.

Lim, H., & Kim, D. (2021). Pairwise dependence-based unsupervised feature selection. *Pattern Recognition*, *111*, Article 107663.

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

Lin, Y., Liu, H., Zhao, H., Hu, Q., Zhu, X., & Wu, X. (2022). Hierarchical feature selection based on label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, *35*, 5964–5976.

Liu, Y., Chen, H., Li, T., & Li, W. (2023). A robust graph based multi-label feature selection considering feature-label dependency. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *53*(1), 837–863.

Liu, T., Hu, R., & Zhu, Y. (2023). Completed sample correlations and feature dependency-based unsupervised feature selection. *Multimedia Tools and Applications*, *82*(10), 15305–15326.

Liu, H., Lin, Y., Liu, J., et al. (2020). Hierarchical feature selection from coarse to fine. *Acta Electronica Sinica*, *50*(11), 2778–2789.

Liu, H., Lin, Y., Wang, C., Guo, L., & Chen, J. (2023). Semantic-gap-oriented feature selection in hierarchical classification learning. *Information Sciences*, *642*, Article 119241.

Liu, J., Sheng, Y., Lan, W., Guo, R., Wang, Y., & Wang, J. (2020). Improved ASD classification using dynamic functional connectivity and multi-task feature selection. *Pattern Recognition Letters*, *138*, 82–87.

Liu, X., & Zhao, H. (2021). Robust hierarchical feature selection with a capped l_2-norm. *Neurocomputing*, *443*, 131–146.

Liu, X., Zhou, Y., & Zhao, H. (2021). Robust hierarchical feature selection driven by data and knowledge. *Information Sciences*, *551*, 341–357.

Qian, W., Xiong, Y., Yang, J., & Shu, W. (2022). Feature selection for label distribution learning via feature similarity and label correlation. *Information Sciences*, *582*, 38–59.

Qiu, Z., & Zhao, H. (2022). A fuzzy rough set approach to hierarchical feature selection based on hausdorff distance. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *52*(10), 11089–11102.

Samareh Jahani, M., Saberi Movahed, F., Eftekhari, M., Aghamollaei, G., & Tiwari, P. (2024). Low-redundant unsupervised feature selection based on data structure learning and feature orthogonalization. *Expert Systems with Applications*, *240*, Article 122556.

Schieber, B., & Vishkin, U. (1988). On finding lowest common ancestors: Simplification and parallelization. *SIAM Journal on Computing*, *17*(6), 1253–1262.

Shao, W., Peng, Y., Zu, C., Wang, M., & Zhang, D. (2020). Hypergraph based multi-task feature selection for multimodal classification of Alzheimer's disease. *Computerized Medical Imaging and Graphics*, *80*, Article 101663.

Shi, J., Li, Z., & Zhao, H. (2023). Feature selection via maximizing inter-class independence and minimizing intra-class redundancy for hierarchical classification. *Information Sciences*, *626*, 1–18.

Shi, J., & Zhao, H. (2023). FS-MGKC: Feature selection based on structural manifold learning with multi-granularity knowledge coordination. *Information Sciences*, *648*, Article 119555.

Tuo, Q., Zhao, H., & Hu, Q. (2019). Hierarchical feature selection with subtree based graph regularization. *Knowledge-Based Systems*, *163*, 996–1008.

Van de Ven, G. M., Tuytelaars, T., & Tolias, A. S. (2022). Three types of incremental learning. *Nature Machine Intelligence*, *4*(12), 1185–1197.

Wang, H., Zhang, P., Zhu, X., Tsang, I., Chen, L., Zhang, C., et al. (2017). Incremental subgraph feature selection for graph classification. *IEEE Transactions on Knowledge and Data Engineering*, *29*(1), 128–142.

Wei, L., Liao, M., Gao, X., & Zou, Q. (2014). An improved protein structural classes prediction method by incorporating both sequence and structure information. *IEEE Transactions on Nanobioscience*, *14*(4), 339–349.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 3485–3492).

Zhang, J., Luo, Z., Li, C., Zhou, C., & Li, S. (2019). Manifold regularized discriminative feature selection for multi-label learning. *Pattern Recognition*, *95*, 136–150.

Zhao, H., Hu, Q., Zhu, P., Wang, Y., & Wang, P. (2021). A recursive regularization based feature selection framework for hierarchical classification. *IEEE Transactions on Knowledge and Data Engineering*, *33*(7), 2833–2846.

Zhao, J., Peng, Y., & He, X. (2020). Attribute hierarchy based multi-task learning for fine-grained image classification. *Neurocomputing*, *395*, 150–159.

Zhao, H., & Yu, S. (2019). Cost-sensitive feature selection via the $l_{2,1}$-norm. *International Journal of Approximate Reasoning*, *104*, 25–37.