Hierarchical Feature Selection with Recursive Regularization

Hong Zhao[1,2], **Pengfei Zhu**[1], **Ping Wang**[1], **Qinghua Hu**[1*]
[1]Tianjin University, China
[2]Lab of Granular Computing, Minnan Normal University, China
{hongzhaocn, zhupengfei, wang_ping, huqinghua}@tju.edu.cn

Abstract

In the big data era, the sizes of datasets have increased dramatically in terms of the number of samples, features, and classes. In particular, there exists usually a hierarchical structure among the classes. This kind of task is called hierarchical classification. Various algorithms have been developed to select informative features for flat classification. However, these algorithms ignore the semantic hyponymy in the directory of hierarchical classes, and select a uniform subset of the features for all classes. In this paper, we propose a new technique for hierarchical feature selection based on recursive regularization. This algorithm takes the hierarchical information of the class structure into account. As opposed to flat feature selection, we select different feature subsets for each node in a hierarchical tree structure using the parent-children relationships and the sibling relationships for hierarchical regularization. By imposing $\ell_{2,1}$-norm regularization to different parts of the hierarchical classes, we can learn a sparse matrix for the feature ranking of each node. Extensive experiments on public datasets demonstrate the effectiveness of the proposed algorithm.

1 Introduction

In the big data era, we are often confronted with classification tasks involving hundreds of classes, where there is a hierarchical structure among the classes. We call this kind of task hierarchical classification. Many real-world classification problems can be naturally cast as hierarchical classification [Silla and Freitas, 2011]. For example, ImageNet [Deng *et al.*, 2009] is an image database organized according to the WordNet [Miller, 1995] hierarchy. These tasks become challenging when the number of classes is very large and testing against every possible class may become computationally infeasible [Bengio *et al.*, 2010]. The hierarchical class structure is important side information for classification learning. Growing attention has been given to structured or hierarchical classification learning in recent years. Learning algorithms that exploit hierarchies have been developed for activities including lung disease classification, text categorization, visual categorization, gene function prediction, and plant species identification [Gopal and Yang, 2015].

With the growth of big data, feature selection [Tang and Liu, 2014; Villela *et al.*, 2015; Wang and Guo, 2017] has received much attention in machine learning. It aims to select a subset of features from the original data to obtain a compact representation of the classification task [Yang *et al.*, 2011]. These feature selection algorithms assume that the classes are independent of each other. In addition, they search for a single feature subset to generate a classifier. However, it is known that some features are useful for distinguishing some classes, but useless for others [Freeman *et al.*, 2013]. Thus we should select different features for different subtasks to construct an appropriate feature subset that leads to a compact and effective classification model.

Obviously, hierarchical class information is not only beneficial for training hierarchical classification models, but is also helpful for selecting a feature subset for each node. However, little work has been devoted to this problem. In hierarchical feature selection, we divide a large-scale classification task into a set of smaller classification problems, where each subtask uses an independent feature subset [Freeman *et al.*, 2013]. Freeman et al. [Freeman *et al.*, 2011] developed a method for joint feature selection and hierarchical classifier design using genetic algorithms. Song et al. [Song *et al.*, 2015] then proposed a feature selection algorithm for hierarchical text classification. However, they did not consider the dependence between different classes in the hierarchical tree, and independently selected features for each node. Classes in a hierarchical structure have both parent-children relationships and sibling relationships. Classes with a parent-children relationship are similar to each other and may share common features for classification, while distinguishing between classes with a sibling relationship may require different features. However, these algorithms evaluate the importance of features individually.

In this paper, we design a hierarchical feature selection method based on recursive regularization. This algorithm considers the hierarchical information of the class structure. First, we model the hierarchical information for parent-children class relationship as a hierarchical regularization. We then use the Hilbert-Schmidt Independence Criterion (H-

*(Corresponding author: Qinghua Hu).

SIC) [Gretton *et al.*, 2005] to measure the independence of the sibling classes, and penalize the dependence between the features selected at sibling nodes. Thus the final subsets are similar if the nodes have a parent-children relationship, while they are different if there is sibling relationship. The contributions of this paper are summarized as follows.

- We first attempt to conduct hierarchical feature selection by considering the hierarchical class structure of parent-children and sibling relationships. These relationships are modeled by hierarchical recursive regularization, which is more reasonable for representing the relationships between nodes than flat approaches.
- In contrast to existing flat algorithms, we model hierarchical feature selection as a convex objective function and explore an alternation minimization strategy to solve the optimization problem with guaranteed convergence.
- Extensive experiments on six hierarchical datasets demonstrate the effectiveness of our approach in terms of efficiency and accuracy.

2 The Proposed Model

In this section, we describe the hierarchical feature selection model with recursive regularization for tasks with hierarchical tree structure.

2.1 Hierarchical Tree Structure

A hierarchical tree is defined as a pair $(\mathcal{D},\prec)$, where $\mathcal{D} = \{1, 2, \cdots\}$ is the set of all classes and "$\prec$" represents the "IS-A" relationship, which is the *subclass-of* relationship with the following properties [Kosmopoulos *et al.*, 2015]:

(1) Asymmetry: if $i \prec j$ then $j \not\prec i$ for every $i, j \in \mathcal{D}$.
(2) Anti-reflexivity: $i \not\prec i$ for every $i \in \mathcal{D}$.
(3) Transitivity: if $i \prec j$ and $j \prec k$, then $i \prec k$ for every $i, j, k \in \mathcal{D}$.

In a hierarchical tree structure,

(1) p_i is the parent of node $i \in \mathcal{D}$;
(2) S_i is the set of all siblings of node $i \in \mathcal{D}$, and $|S_i|$ is the number of the siblings of i;
(3) C_i is the set of all children of node $i \in \mathcal{D}$, and $|C_i|$ is the number of the children of i.

Table 1 describes the most frequent symbols used throughout this paper.

Table 1: Description of symbols used throughout the article.

Symbol	Meaning
p_i	The parent category of class i
C_i	The set of child categories of class i
S_i	The set of sibling categories of class i
$\|C_i\|$	The number of child categories of class i
$\|S_i\|$	The number of sibling categories of class i

2.2 Model

Let $\mathbf{X}_i \in \mathbb{R}^{m_i \times n}$ be a data matrix, where m_i is the number of the samples in subtree of node i, and n is the numbers of features. We use $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{m_i}$ to represent the m_i samples, $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{X}_i = [\mathbf{x}_1; \mathbf{x}_2; \cdots; \mathbf{x}_{m_i}]$. Let $\mathbf{Y}_i \in \mathbb{R}^{m_i \times d_{max}}$ be a class matrix, where d_{max} is the largest number of sub-classes. We use $\mathbf{y}_i \in \{0, 1\}^{d_{max}}$ to represent the class of sample $\mathbf{y}_i$, and $\mathbf{Y}_i = [\mathbf{y}_1; \mathbf{y}_2; \cdots; \mathbf{y}_{m_i}]$.

Let $\|\cdot\|_F$ denote the Frobenius norm of a matrix, and let $\|\cdot\|_{2,1}$ denote the $\ell_{2,1}$-norm of a matrix. In the context of hierarchies, the primary optimization problem is estimating the parameters $\mathbf{W}_i$ at each node:

$$J = \min_{\mathbf{W}_i} \sum_{i=0}^{N} (\|\mathbf{X}_i\mathbf{W}_i - \mathbf{Y}_i\|_F^2 + \lambda\|\mathbf{W}_i\|_{2,1}), \tag{1}$$

where $\mathbf{W}_i \in \mathbb{R}^{n \times d}$ is the feature weight matrix, the first term is the loss item, the second term is the regularization imposed on $\mathbf{W}_i$, λ is a positive constant, and N is the number of internal nodes.

Example 1 *An example of a hierarchical class tree structure is shown in Figure 1. From this figure, we have*

(1) The parent category of class 1 *is* $p_1 = 0$*;*

(2) The set of child categories of class 0 *is* $C_0 = \{1, 2\}$*;*

(3) The set of sibling categories of class 3 *is* $S_3 = \{4\}$.

Hierarchical feature selection should compute the feature weight matrix $\mathbf{W}_i$ *for each node besides leaf nodes.*

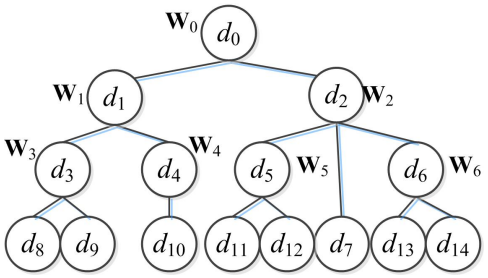


Figure 1: Tree structure ($h = 4$).

In the hierarchical class structure, there are parent-children relationship and sibling relationship. We impose these two kinds of relationship as regularization terms on $\mathbf{W}$ to select features.

Classes have a parent-children relationship means that they are neighboring nodes among all hierarchical classes. We expect that they are similar to each other and should share common features for classification. We first introduce this relationship in the hierarchy into the learning process by incorporating a recursive structure into the regularization term. The regularization term of parent-children relationship is

$$\sum_{i=1}^{N} \|\mathbf{W}_i - \mathbf{W}_{p_i}\|_F^2. \tag{2}$$

In addition, we expect the features at different sibling nodes to be different from each other. For example, the textural features can identify animals, but the edge features are representative for furniture. We measure the independence using

a kernel dependence measure (the HSIC) by mapping variables into a reproducing kernel Hilbert space (RKHS). This criterion measures the high-order joint moments between the original distributions [Bach and Jordan, 2002]. We use the HSIC to penalize the dependence between the selected features at sibling nodes in an RKHS.

Let $\mathbf{K}_i$ and $\mathbf{K}_l$ be kernel spaces on $\mathbf{W}_i \in \mathbb{R}^{n\times d}$ and $\mathbf{W}_l \in \mathbb{R}^{n\times d}$, where $l \in S_i$ is the l-th sibling node of node i, and $\mathbf{W}_i$ and $\mathbf{W}_l$ are the representation coefficient matrices for the i-th node and the l-th node, respectively. Then

$$HSIC(\mathbf{W}_i, \mathbf{W}_l) = tr(\mathbf{K}_i\mathbf{H}\mathbf{K}_l\mathbf{H}), \tag{3}$$

where $\mathbf{K}_i = \mathbf{W}_i\mathbf{W}_i^T$, $\mathbf{K}_l = \mathbf{W}_l\mathbf{W}_l^T$, and $1 \le l \le |S_i|$. $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \in \mathbb{R}^{n\times n}$ centers the matrix to have zero mean, where $\mathbf{1}_n \in \mathbb{R}^n$ is a column vector with all elements being 1.

In the context of hierarchies, the primary optimization problem considering the parent-children and sibling relationships is formulated as

$$\begin{aligned} J = \min_{\mathbf{W}_i} &\sum_{i=0}^{N}(\|\mathbf{X}_i\mathbf{W}_i - \mathbf{Y}_i\|_F^2 + \lambda\|\mathbf{W}_i\|_{2,1}) \\ &+ \alpha\sum_{i=1}^{N}\|\mathbf{W}_i - \mathbf{W}_{p_i}\|_F^2 + \beta\sum_{i=1}^{N}\sum_{l\in S_i} HSIC(\mathbf{W}_i, \mathbf{W}_l), \end{aligned} \tag{4}$$

where $i = 0$ indicates a root node with no parent node or sibling nodes. Therefore, the value of i in the two regularization terms starts at 1. We call this task hierarchical feature selection with recursive regularization (HiFSRR).

2.3 Optimization of HiFSRR

Because of the non-smoothness of the $\ell_{2,1}$-norm, it is difficult to derive a closed solution to the optimization problem in Eq. (4) directly. According to [Nie *et al.*, 2010], this problem can be solved in an alternative way. When $w_i \neq 0$ for $i = 1, \cdots, d$, the derivative of $\|\mathbf{W}\|_{2,1}$ with respect to $\mathbf{W}$ is

$$\frac{\partial(\|\mathbf{W}\|_{2,1})}{\partial\mathbf{W}} = 2\mathbf{D}\mathbf{W}, \tag{5}$$

where $\mathbf{D} \in \mathbb{R}^{d\times d}$ is a diagonal matrix with the i-th diagonal element as $\mathbf{D}_{jj} = \frac{1}{2\|w_j\|_2}$. It can be easily verified that the derivative in Eq. (5) can also be regarded as the derivative of $Tr(\mathbf{W}^T\mathbf{D}\mathbf{W})$. Thus, the optimization problem is written as

$$\begin{aligned} J = \min_{\mathbf{W}_i} &\sum_{i=0}^{N}(\|\mathbf{X}_i\mathbf{W}_i - \mathbf{Y}_i\|_F^2 + \lambda Tr(\mathbf{W}_i^T\mathbf{D}_i\mathbf{W}_i)) \\ &+ \alpha\sum_{i=1}^{N}\|\mathbf{W}_i - \mathbf{W}_{p_i}\|_F^2 + \beta\sum_{i=1}^{N}\sum_{l\in S_i} HSIC(\mathbf{W}_i, \mathbf{W}_l). \end{aligned} \tag{6}$$

The root node should be computed individually. Therefore,

Algorithm 1 Hierarchical Feature Selection with Recursive Regularization (HiFSRR)

Input: Input data $\mathbf{X}_i \in \mathbb{R}^{m_i\times n}$ and labels $\mathbf{Y}_i \in \{0,1\}^{m_i\times d_{max}}$, where $i = 0, 1, \cdots, N$, and N is the number of internal nodes. To facilitate the calculation, we let d_{max} be the maximum number of classes of internal nodes. Regularization parameters λ, α, and β.
Output: Matrix $\mathbf{W} \in \mathbb{R}^{n\times d_{max}(N+1)}$.

```
Set t = 0 and initialize W_i ∈ R^{n×d_max} randomly;
W = [W_0, W_1, ··· , W_N];
repeat
    for i = 0 : N do
        Compute the diagonal matrix D_i^(t) according to d^i_jj = 1/(2||w^i_j||_2);
    end for
    // Update the root node.
    Update W_0 by W_0^(t+1) = (X_0^T X_0 + λD_0^(t) + α|C_0|I)^{-1}(X_0^T Y_0 + α Σ_{i∈C_0} W_i^(t));
    // Update the internal nodes.
    for i = 1 : N do
        Update W_i by W_i^(t+1) = (X_i^T X_i + λD_i^(t) + αI + β Σ_{l∈S_i}(U_l + U_l^T))^{-1}(X_i^T Y_i + αW_{p_i}^(t));
    end for
    Update W^(t+1) = [W_0, W_1, ··· , W_N];
    t = t + 1;
until Convergence criterion satisfied;
return W;
```

the objective function is rewritten as

$$\begin{aligned} J = \min_{\mathbf{W}_0,\mathbf{W}_i} &\|\mathbf{X}_0\mathbf{W}_0 - \mathbf{Y}_0\|_F^2 + \lambda Tr(\mathbf{W}_0^T\mathbf{D}_0\mathbf{W}_0) \\ &+ \sum_{i=1}^{N}(\|\mathbf{X}_i\mathbf{W}_i - \mathbf{Y}_i\|_F^2 + \lambda Tr(\mathbf{W}_i^T\mathbf{D}_i\mathbf{W}_i)) + \alpha\|\mathbf{W}_i - \mathbf{W}_{p_i}\|_F^2 \\ &+ \beta\sum_{i=1}^{N}\sum_{l\in S_i} Tr(\mathbf{W}_i\mathbf{W}_i^T\mathbf{H}\mathbf{W}_l\mathbf{W}_l^T\mathbf{H}). \end{aligned} \tag{7}$$

For the root of the tree, by setting the derivative of Eq. (7) w.r.t. $\mathbf{W}_0$ to 0, we have

$$\begin{aligned} \frac{\partial J}{\partial\mathbf{W}_0} &= 2\mathbf{X}_0^T(\mathbf{X}_0\mathbf{W}_0 - \mathbf{Y}_0) + 2\lambda\mathbf{D}_0\mathbf{W}_0 - 2\alpha\sum_{i\in C_0}(\mathbf{W}_i - \mathbf{W}_0) \\ &= (2\mathbf{X}_0^T\mathbf{X}_0 + 2\lambda\mathbf{D}_0 + 2\alpha|C_0|\mathbf{I})\mathbf{W}_0 - 2\mathbf{X}_0^T\mathbf{Y}_0 - 2\alpha\sum_{i\in C_0}\mathbf{W}_i \\ &= 0, \end{aligned} \tag{8}$$

where i is the i-th child of root node C_0, and $|C_0|$ is the number of all children of root node. Therefore, we have

$$\mathbf{W}_0 = (\mathbf{X}_0^T\mathbf{X}_0 + \lambda\mathbf{D}_0 + \alpha|C_0|\mathbf{I})^{-1}(\mathbf{X}_0^T\mathbf{Y}_0 + \alpha\sum_{i\in C_0}\mathbf{W}_i). \tag{9}$$

Let $\mathbf{U}_l = \mathbf{H}\mathbf{W}_l\mathbf{W}_l^T\mathbf{H}$. By setting the derivative of Eq. (7) w.r.t. internal node $\mathbf{W}_i$ to 0, we have

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{W}_i} &= 2\mathbf{X}_i^T(\mathbf{X}_i\mathbf{W}_i - \mathbf{Y}_i) + 2\lambda\mathbf{D}_i\mathbf{W}_i + 2\alpha(\mathbf{W}_i - \mathbf{W}_{p_i}) \\ &\quad + \beta\sum_{l\in S_i}(\mathbf{U}_l + \mathbf{U}_l^T)\mathbf{W}_i \\ &= (2\mathbf{X}_i^T\mathbf{X}_i + 2\lambda\mathbf{D}_i + 2\alpha\mathbf{I} + \beta\sum_{l\in S_i}(\mathbf{U}_l + \mathbf{U}_l^T))\mathbf{W}_i \\ &\quad - 2\mathbf{X}_i^T\mathbf{Y}_i - 2\alpha\mathbf{W}_{p_i} \\ &= 0.\end{aligned} \tag{10}$$

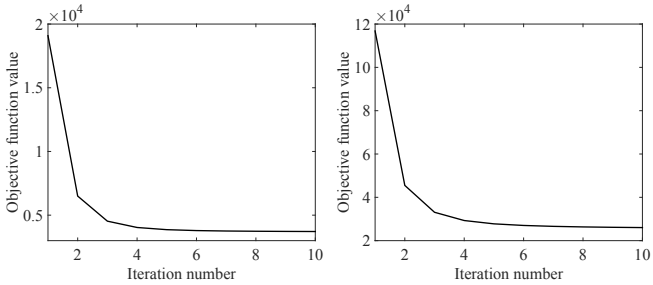
Finally, we have

$$\mathbf{W}_i = (\mathbf{X}_i^T\mathbf{X}_i + \lambda\mathbf{D}_i + \alpha\mathbf{I} + \beta\sum_{l\in S_i}(\mathbf{U}_l + \mathbf{U}_l^T))^{-1}(\mathbf{X}_i^T\mathbf{Y}_i + \alpha\mathbf{W}_{p_i}). \tag{11}$$

The HiFSRR algorithm is formulated in Algorithm 1. With this algorithm, the weight vector $\mathbf{W} = [\mathbf{W}_0, \mathbf{W}_1, \cdots, \mathbf{W}_N]$ is obtained. We sort the n features for the i-th node according to $\|w_{ij}\|_F (j = 1, \cdots, n)$ in the descending order and select the top ranked subsets at this node, where $i = 0, \cdots, N$ and N is the number of internal nodes.

2.4 Convergence Analysis

Algorithm 1 monotonically decreases the value of the objective function for the problem in Eq. (6) in each iteration, and converges to the global optimum of the problem. The $\ell_{2,1}$-norm minimization problem has been studied and the convergence has been proved in [Nie *et al.*, 2010]. The following experiments also show that the proposed algorithm converges quickly on Cifar and SUN datasets. We find the same phenomenon on other datasets.



(a) (b)

Figure 2: Convergence curves of the objective function value. (a) *Cifar*; (b) *SUN*.

Figure 2 shows the convergence curves of all the datasets, which shows that the objective function value decreases monotonically and converges within no more than ten iterations for all datasets.

3 Experiments

In this section, we introduce the datasets used in our experiments. We then analyze the parameter sensitivity and report effectiveness of regularization terms. Finally, we compare the proposed algorithm with some state of the art techniques.

3.1 Datasets

The experiments use protein and image datasets to test the proposed algorithm. All these tasks have the information of class hierarchy. Two protein tasks include: F194 [Wei *et al.*, 2015] and DD [Ding and Dubchak, 2001]. Four image tasks include: CLEF [Dimitrovski *et al.*, 2011], CIFAR-100 [Krizhevsky and Hinton, 2009], PASCAL Visual Object Classes (VOC) [Everingham *et al.*, 2010], and Scene UNderstanding (SUN) [Xiao *et al.*, 2010]. There are multi-label objects in the SUN dataset. As we do not discuss this kind of tasks in this work, we remove these multi-label samples. A description of the datasets is given in Table 2.

Table 2: Data description.

No.	Dataset	Train	Test	Feature	Node	Leaf	Height
1	F194	7105	1420	473	202	194	3
2	DD	3020	605	473	32	27	3
3	CLEF	8368	939	80	88	80	4
4	Cifar	50000	10000	512	121	100	3
5	VOC	7178	5105	1000	30	20	5
6	SUN	45109	22556	4096	343	324	4

3.2 Comparison Methods

So far, little research has been devoted to developing feature selection algorithms for hierarchical classification. In [Freeman *et al.*, 2011], only two toy hierarchical datasets were presented. In [Song *et al.*, 2015], the algorithm was designed specifically for hierarchical text datasets. Therefore, we compare HiFSRR with the following flat feature selection algorithms.

(1) **Fisher Score**: which depends on fully labeled training data to select features with the best discriminating ability [Duda *et al.*, 2012].

(2) **FSNM**: Feature Selection via Joint $\ell_{2,1}$-norms Minimization [Nie *et al.*, 2010] which employs joint $\ell_{2,1}$-norm minimization on both loss function and regularization to realize feature selection across all data points.

(3) **mRMR**: Minimal-redundancy-maximal-relevance criterion (mRMR) is an effective feature selection scheme which avoids the difficult multivariate density estimation in maximizing dependency [Peng *et al.*, 2005].

(4) **Relief**: Relief is a classical feature selection algorithm inspired by instance-based learning [Kira and Rendell, 1992].

3.3 Parameter Setting

Features are selected individually for each internal class node by using the parent-children and sibling relationships simultaneously. For example, Figure 3 shows the class hierarchical tree for the VOC dataset. We select a feature subset for the nodes *Objects*, *Vehicles*, *Household*, and *Animals*. We select features on training sets and test them on test sets using 10-fold cross validation. We report the average accuracy of SVM at internal nodes.

In the experiments, we set $\lambda = 1$, $\beta = 1$, and $\alpha = 1$ for the CLEF dataset, and set $\lambda = 10$, $\beta = 0.1$, and $\alpha = 0.1$ for the

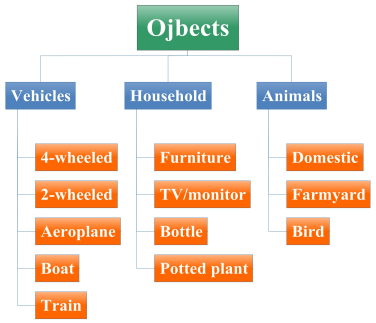
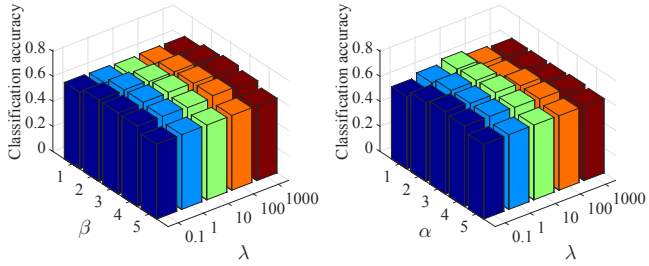


Figure 3: Class hierarchical tree of VOC dataset.

other datasets. We set the number of selected features to be {48, 40, 32, 24} for F194 and DD datasets. We set the number of selected features to be {400, 300, 200, 100} and {200, 160, 120, 80, 40} for the VOC and SUN datasets, respectively. We set the number of selected features to be {256, 205, 154, 103, 52} for the Cifar dataset which are 50%, 40%, 30%, 20%, and 10% of the features.



(a) (b)

Figure 4: Parameter sensitivity evaluation on *Vehicles* node of VOC dataset.

We analyze the parameter sensitivity, as shown in Figure 4. We fix the value of one parameter (with 1) and tune the others. Due to the space limit, we report the results on *Vehicles* node of VOC dataset with fixed α and β, respectively. The results show that our method is not sensitive to parameters.

3.4 Effectiveness of Regularization Terms

We investigate experimentally the performance of regularization terms in our HiFSRR. We compare the effectiveness of HiFSRR with two parameter settings using different datasets.

(1) $\alpha = 1$ and $\beta = 1$ mean that HiFSRR with hierarchical relationships.

(2) $\alpha = 0$ and $\beta = 0$ mean that HiFSRR without hierarchical relationships.

Figure 5 shows the average accuracy of SVM on each node of a dataset. The results demonstrate that the regularization terms of parent-children and sibling relationships work well with different numbers of selected features. The advantage of two regularization terms is most obvious when small features are selected.

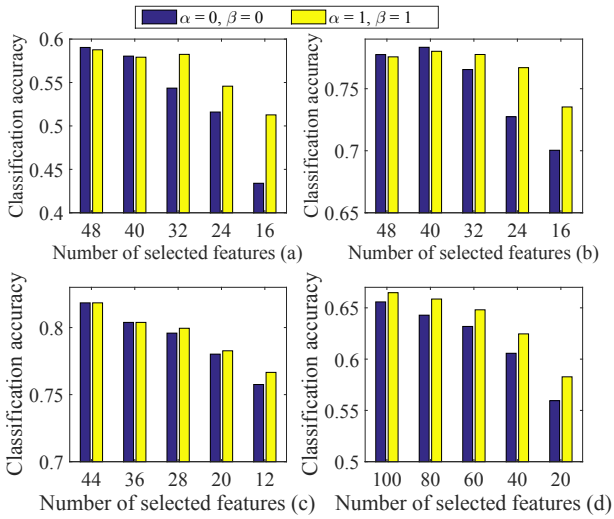


Figure 5: Accuracy of SVM on different nodes of VOC dataset with different numbers of selected features. (a) *Objects*; (b) *Vehicles*; (c) *Household*; (d) *Animals*.

3.5 Experimental Results and Discussion

First, we compare the efficiency of the four flat feature selection algorithms and the HiFSRR algorithm according to running time. We then compare the performance of different algorithms that are dominated by SVM. Finally, we use two large datasets, Cifar and SUN, to test the performance of our method on large datasets.

All experiments are executed on an Intel Core i7-3770 running at 3.40 GHz with 12.0 GB memory and 64-bit Windows 7 operating system. The results presented in Table 3 demonstrate that the HiFSRR algorithm has a significantly shorter running time than the other algorithms except for the Fisher algorithm. Neither FSNM nor Relief can process the Cifar and SUN datasets with running out of memory.

Table 3: Running time (s).

Dataset	Fisher	FSNM	mRMR	Relief	HiFSRR
F194	5.4	154.2	84.0	241.0	5.9
DD	1.0	16.0	45.6	61.8	1.9
CLEF	0.2	222.3	2.0	43.9	0.2
VOC	1.4	81.3	194.3	341.4	7.6
Cifar	6.0	–	391.0	–	50.1
SUN	118.1	–	34620.3	–	3529.2

We compare HiFSRR with the four flat feature selection algorithms on two protein datasets and two image datasets.

Results on protein datasets. Comparisons of the feature selection algorithms for the F194 and DD datasets are shown in Table 4. We select a feature subset for each internal node of the class hierarchical tree. The results in Table 4 give the average accuracy at each internal node using SVM. It is clear that, in most cases, HiFSRR performs better than other approaches given a different number features, especially when small

Table 4: Accuracy comparison of SVM on two protein datasets with different numbers of selected features.

Dataset	Method	48	40	32	24
F194	Fisher	0.5295	0.4950	0.4608	0.4114
	FSNM	0.5750	0.5494	0.5388	0.5134
	mRMR	**0.5773**	0.5619	0.5475	0.5111
	Relief	0.4022	0.3747	0.3424	0.3246
	HiFSRR	0.5765	**0.5776**	**0.5614**	**0.5438**
DD	Fisher	0.6905	0.6413	0.6318	0.6146
	FSNM	0.7749	**0.7836**	0.7591	0.7215
	mRMR	0.7787	0.7716	0.7670	0.7388
	Relief	0.6327	0.6301	0.6116	0.6057
	HiFSRR	**0.7791**	0.7748	**0.7788**	**0.7791**

feature subsets are selected. For example, the classification of the DD dataset with 24 features achieves good accuracy.

Results on CLEF dataset. A comparison of the feature selection algorithms for the CLEF dataset is shown in Table 5. The results demonstrate that the classification results for different internal nodes using the HiFSRR algorithm with 40 features are generally better than those using the flat feature selection algorithms except for Relief.

Table 5: Accuracy comparison of SVM on different internal node of CLEF dataset.

	Fisher	FSNM	mRMR	Relief	HiFSRR
Node 1	0.7498	0.7838	0.7626	0.7710	**0.8096**
Node 2	0.8869	0.8173	0.9008	**0.9567**	0.8586
Node 3	0.6754	0.7104	0.7029	0.6860	**0.7645**
Node 4	0.6552	0.7463	0.7043	0.6792	**0.7678**
Node 5	0.7019	0.6887	0.6304	0.6893	**0.7062**
Node 6	0.9775	0.9775	0.9674	**0.9876**	0.9825
Average	0.7745	0.7873	0.7781	0.7950	**0.8149**

Results on VOC dataset. The experimental results on VOC dataset are shown in Figure 6. We can see from Figure 6 that the classification results for different internal nodes using the HiFSRR algorithm are in general better than those using the flat feature selection algorithms. The advantage of our algorithm is most obvious when 100 features are selected.

Results on large datasets. Finally, we test our algorithm on two large datasets. Tables 6(a) and 6(b) compare the results of the performance of the HiFSRR algorithm with the Fisher algorithm and mRMR on the Cifar and SUN datasets, respectively. The results demonstrate that HiFSRR algorithm obtains superior performance in different feature subsets.

Moreover, the number of features is significantly reduced by HiFSRR, leading to much faster classification, especially for large datasets.

4 Conclusions and Future Work

We have proposed a hierarchical feature selection approach based on recursive regularization to exploiting the parent-children and sibling relationships of hierarchical class structures. In contrast to existing feature selection approaches, we take advantage of the hierarchical class structure, which provides significant information for classification learning. We have also provided an efficient HiFSRR algorithm to select different feature subsets for each node in a hierarchical tree structure. Compared with the flat feature selection approach, HiFSRR achieves competitive results for both classification accuracy and computational efficiency. The current implementation of the algorithm only deals with a tree structure for class labels in which case each node (class) has a single parent. In the future, we will design feature selection approaches for graph structures which is more general than tree structure.

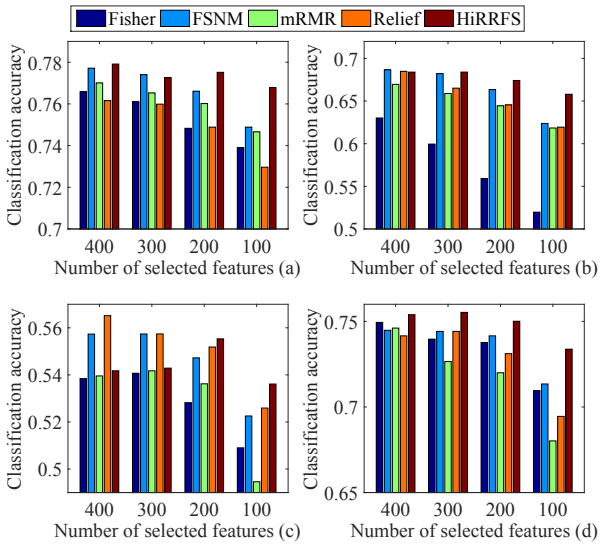


Figure 6: Accuracy of SVM on different nodes of VOC dataset with different numbers of selected features. (a) *Objects*; (b) *Vehicles*; (c) *Household*; (d) *Animals*.

Table 6: Accuracy of SVM on two large datasets with different numbers of selected features.

(a) *Cifar*

	256	205	154	103	52
Firsher	0.5092	0.5030	0.4882	0.4707	0.4272
mRMR	0.5096	0.5043	0.4881	0.4689	0.4107
HiFSRR	**0.5244**	**0.5185**	**0.5148**	**0.5079**	**0.4767**

(b) *SUN*

	200	160	120	80	40
Fisher	0.7176	0.6987	0.6657	0.6069	0.4889
mRMR	0.7220	0.7020	0.6711	0.6197	0.5103
HiFSRR	**0.7343**	**0.7191**	**0.6865**	**0.6391**	**0.5346**

Acknowledgments

This work was supported by the National Program on Key Basic Research Project under Grant 2013CB329304, and the National Natural Science Foundation of China under Grant Nos. 61432011, U1435212, 61502332, and 61379049.

References

[Bach and Jordan, 2002] Francis R Bach and Michael I Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3(Jul):1–48, 2002.

[Bengio *et al.*, 2010] Samy Bengio, Jason Weston, and David Grangier. Label embedding trees for large multi-class tasks. In *Advances in Neural Information Processing Systems*, pages 163–171, 2010.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li Jia Li, Kai Li, and Li Fei Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[Dimitrovski *et al.*, 2011] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski. Hierarchical annotation of medical images. *Pattern Recognition*, 44(10):2436–2449, 2011.

[Ding and Dubchak, 2001] Chris HQ Ding and Inna Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4):349–358, 2001.

[Duda *et al.*, 2012] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[Everingham *et al.*, 2010] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[Freeman *et al.*, 2011] Cecille Freeman, Dana Kulić, and Otman Basir. Joint feature selection and hierarchical classifier design. In *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1728–1734. IEEE, 2011.

[Freeman *et al.*, 2013] C Freeman, D Kuli, and O Basir. Feature-selected tree-based classification. *IEEE Transactions on Cybernetics*, 43(6):1990–2004, 2013.

[Gopal and Yang, 2015] Siddharth Gopal and Yi Ming Yang. Hierarchical bayesian inference and recursive regularization for large-scale classification. *ACM Transactions on Knowledge Discovery from Data*, 9(3):18, 2015.

[Gretton *et al.*, 2005] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International Conference on Algorithmic Learning Theory*, pages 63–77. Springer, 2005.

[Kira and Rendell, 1992] Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Workshop on Machine Learning*, pages 249–256, 1992.

[Kosmopoulos *et al.*, 2015] Aris Kosmopoulos, Ioannis Partalas, Eric Gaussier, Georgios Paliouras, and Ion Androutsopoulos. Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29(3):820–865, 2015.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[Miller, 1995] George A Miller. Wordnet: a lexical database for english. *Communications of the Acm*, 38(11):39–41, 1995.

[Nie *et al.*, 2010] Fei Ping Nie, Heng Huang, Xiao Cai, and Chris H Ding. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *Advances in Neural Information Processing Systems*, pages 1813–1821, 2010.

[Peng *et al.*, 2005] Han Chuan Peng, Fu Hui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[Silla and Freitas, 2011] Carlos N. Silla and Alex A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72, 2011.

[Song *et al.*, 2015] Jia Song, Pengzhou Zhang, Sijun Qin, and Junpeng Gong. A method of the feature selection in hierarchical text classification based on the category discrimination and position information. In *International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration*, pages 132–135, 2015.

[Tang and Liu, 2014] Jiliang Tang and Huan Liu. An unsupervised feature selection framework for social media data. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2914–2927, 2014.

[Villela *et al.*, 2015] Saulo Moraes Villela, De Castro Leite Saul, and Raul Fonseca Neto. Feature selection from microarray data via an ordered search with projected margin. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 3874–3881, 2015.

[Wang and Guo, 2017] Shi Ping Wang and Wen Zhong Guo. Sparse multi-graph embedding for multimodal feature representation. *IEEE Transactions on Multimedia*, 2017.

[Wei *et al.*, 2015] Leyi Wei, Minghong Liao, Xing Gao, and Quan Zou. An improved protein structural classes prediction method by incorporating both sequence and structure information. *IEEE Transactions on Nanobioscience*, 14(4):339–349, 2015.

[Xiao *et al.*, 2010] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492. IEEE, 2010.

[Yang *et al.*, 2011] Yi Yang, Heng Tao Shen, Zhi Gang Ma, Zi Huang, and Xiao Fang Zhou. $\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, volume 22, pages 1589–1594, 2011.