Hierarchical Feature Selection with Orthogonal Transfer

Limei Dong, Hong Zhao

Fujian Key Laboratory of Granular Computing and Application, Minnan Normal University, China
School of Computer Science, Minnan Normal University, China
D_Limei@163.com, hongzhaocn@163.com

Abstract

Feature selection is an indispensable preprocessing step in high-dimensional data classification, which has an effect on both the running time and the result quality of the subsequent classification processing steps. Most existing approaches use flat strategies, which treat each category or class separately and ignore hierarchical structure. In this paper, we propose a hierarchical feature selection algorithm with orthogonal transfer. We first compute the weight of the feature to the category by hierarchical SVM with orthogonal transfer. More specifically, we use an objective that is a convex function of the normal vectors to compute the weight. Then, we select features using the weight and predict the class label for a test sample according to classifier. Finally, extensive experimental results on various real-life datasets have demonstrated the superiority of the proposed algorithm.

Keywords: Hierarchical classification, Feature selection, Orthogonal transfer

1 Introduction

In machine learning, pattern recognition and data mining [1], one is often confronted with multi-class classification problems [2]. Multi-class classification problems have a large number of classes and extremely marvelous number of features [3]. It significantly increases the time and space requirements for processing the data [4]. Classification problems are analytically or computationally manageable in high-dimensional spaces which become completely intractable [5]. Feature selection is an indispensable preprocessing step in high-dimensional data classification, which has effects on both the running time and the result quality of the subsequent processing steps [6].

Feature selection is designed to find the relevant feature subset of the original features which can improve the predictive accuracy of a classification algorithm. Existing feature selection methods focus on "flattening" the class structure, ignoring the hierarchical structure which is popular in many real-world knowledge systems [7]. Although many real world classification problems have complex hierarchical structures such as MeSH, U.S. Patents, Yahoo!, LookSmart and so on, few learning methods capitalize on the structure [8]. Testing every possible class can become computationally infeasible when there are a lot of classes in multi-class classification [9]. In each of these cases, the problem can be alleviated by imposing a hierarchical class structure. The class labels are naturally organized in the form of a hierarchical structure which defines an abstraction over class labels [10].

In this paper, we address open challenges in large-scale classification, focusing on how to effectively leverage the hierarchical structure among class labels. We propose an approach hierarchical feature selection with orthogonal transfer, and deal with it through considering the weight of the feature to the category. One superiority for hierarchical feature selection exploiting the hierarchical structure is to decouple the problem into a set of independent problems. We first compute the weight of the feature to the category in the hierarchy by hierarchical Support Vector Machine (SVM) with orthogonal transfer. Second, we select features utilizing the weight and eliminate redundant features because tree nodes are closely connected in the hierarchy with similar semantic information. It's easy to predict the class label for a test sample employ classifier according to selected features. Extensive experimental results on various real-life datasets have demonstrated the superiority of the proposed algorithm.

The rest of the paper is organized as follows. Section 2 presents feature selection algorithm to address this hierarchical classification problem. In Section 3, we discuss the experimental settings and results. Finally, Section 4, we conclude and suggest further research trend.

2 Model

Let $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\}$ be a set of samples, where each $\mathbf{x}_i \in \mathbf{X}$ and each label y_i refers to a unique category encoded as an integer. Each $y_i \in \mathbf{Y} = \{1, \cdots, q\}$,

*Corresponding Author: Hong Zhao; E-mail: hongzhaocn@163.com
DOI: 10.3966/160792642019072004019

where q is the total number of categories. A tree structured class represents class memberships at different levels of abstraction. The leaves of class indicate the most specific labels. The labels in $\mathbf{Y}$ are identified as leaf nodes in a category tree. Hierarchical SVM encourages the classifiers at each node of the tree to be different from the classifiers at its ancestors. More specifically, it is regularization that force the normal vector of the classifying hyperplane at each node of the tree to be orthogonal to those at its ancestors as much as possible [11].

The goal is to learn a classification function $f:\mathbf{X}\to\mathbf{Y}$ that attains a small classification error. We define some of the symbols used in the text. For each node $v\in\mathbf{Y}$, denote as $\mathbf{A}(v)$ the set of ancestors of v (excluding itself), $\mathbf{S}(v)$ the set of siblings of v, $\mathbf{C}(v)$ the set of children of v, and $\mathbf{D}(v)$ the set of descendants of v (excluding itself). For convenience, we also define $\mathbf{A}^+(v)=\mathbf{A}(v)\cup\{v\}$ and $\mathbf{D}^+(v)=\mathbf{D}(v)\cup\{v\}$. We associate each node $v\in\mathbf{Y}$ with a vector $\mathbf{w}_v\in R^n$, and focus on classifiers $f(x)$ determined by recursive procedure. It is clear that the task of learning $f(x)$ is reduced to learn the set of vectors $\{\mathbf{w}_v \mid v\in\mathbf{Y}\}$, which corresponds to the normal vectors of the classifying hyperplanes.

These regularization terms force each normal vector to be orthogonal to those at its ancestors as much as possible. More specifically, we use the following minimize optimization objective function formulation [12]:

$$\begin{aligned}&\frac{1}{2}\sum_{v\in\mathbf{Y}}k(v,v)\|\mathbf{w}_v\|^2+\sum_{u\in\mathbf{A}(v)}\sum_{v\in\mathbf{Y}(v)}k(u,v)\,|\mathbf{w}_u^T\mathbf{w}_v|\\&\quad+C\sum_{i=1}^{n}\xi_i\\&s.t.\mathbf{w}_v^T\mathbf{x}_i-\mathbf{w}_u^T\mathbf{x}_i\geq 1-\xi_i\\&\forall u\in\mathbf{S}(v),\forall v\in\mathbf{A}^+(y_i),\xi_i\geq 0,\forall i\in\{1,\cdots,n\},\end{aligned}\tag{1}$$

where the optimization variables are the normal vectors $\mathbf{w}_v$ for all $v\in\mathbf{Y}$ and the slack variables ξ_i for all $i\in\{1,\cdots,n\}$. The pairwise function $k:\mathbf{Y}\times\mathbf{Y}\to R_+$ (a nonnegative matrix). The classification margins at different levels in the hierarchy can be effectively differentiated by setting the diagonal coefficients $k(u,v)$. The regularization terms $|\mathbf{w}_u^T\mathbf{w}_v|$ stands for the normal vector of the classifying hyperplane at each node of the tree to be orthogonal to those at its ancestors. We assume k is symmetric, i.e., $k(u,v)=k(v,u)$ for all $u,v\in\mathbf{Y}$. The constant C is parameters that need to be selected before solving the above optimization problem.

The nonnegative pairwise function k such as the formulation Eq. (1) is a convex optimization problem. If the symmetric pairwise function $k:\mathbf{Y}\times\mathbf{Y}\to R_+$ defined by

$$k(u,v)=\begin{cases}\left|D^+(v)\right| & if\ u=v\\ \mu & if\ u\in\mathbf{A}(v)\\ 0 & else\end{cases},\tag{2}$$

where $\mu>0$ is a parameter. For many problems in practice, setting $\mu=1$ often gives a positive definite k, although it is certainly not always true. In any case, we can always reduce the value of μ, or increase the diagonal values $k(v,v)$, to make k positive definite. Then the objective function of Eq. (1) is convex. Therefore it has a unique solution $\mathbf{w}^*$. Establishing convexity of an optimization problem does not always mean that it can be readily solved by existing algorithms and available software.

Considering that the objective function has a constraint, we can find the solution to this optimization problem in dual variables by finding the saddle point of the Lagrangian. However, our Lagrangian formulation has difficulty in solving the difficult problem because of complex parameters. We transform the problem Eq. (1) into an equivalent unconstrained optimization problem

$$\begin{aligned}\underset{\mathbf{w}}{\text{minimize}}\ \theta(\mathbf{w})\cong&\frac{1}{2}\sum_{v\in\mathbf{Y}}k(v,v)\|\mathbf{w}_v\|^2\\&+\sum_{u\in\mathbf{A}(v)}\sum_{v\in\mathbf{Y}(v)}k(u,v)\,|\mathbf{w}_u^\top\mathbf{w}_v|\\&+C\sum_{i=1}^{n}max\{0,\max_{u\in\mathbf{S}(v),v\in\mathbf{A}^+(y_i)}\{1-\mathbf{w}_v^\top\mathbf{x}_i+\mathbf{w}_u^\top\mathbf{x}_i\}\},\end{aligned}\tag{3}$$

where $\mathbf{w}$ is the optimization variable (often called weights in learning problems).

Theorem 1 If the symmetric pairwise function k is positive definite, then the solution to the optimization problem Eq. (5) admits a representation of the form $\mathbf{w}_v=\sum_{i=1}^{n}c_{vi}\mathbf{x}_i$ for any $v\in\mathbf{Y}$.

Here we develop a variant of the regularized dual averaging (RDA) method [13]. We consider regularized stochastic learning and optimization problems, where the objective function is the sum of two convex terms. Let $\lambda_{min}>0$ be its smallest eigenvalue. Then we decompose the objective function in Eq. (5) as $\theta(\mathbf{w})=\phi(\mathbf{w})+\varphi(\mathbf{w})$ into two separate terms:

$$\begin{aligned}\phi(\mathbf{w})=&\frac{1}{2}\sum_{v\in\mathbf{Y}}(k(v,v)-\lambda_{min})\|\mathbf{w}_v\|^2\\&+\sum_{u\in\mathbf{A}(v)}\sum_{v\in\mathbf{Y}(v)}k(u,v)\,|\mathbf{w}_u^\top\mathbf{w}_v|\\&+C\sum_{i=1}^{n}max\{0,\max_{u\in\mathbf{S}(v),v\in\mathbf{A}^+(y_i)}\{1-\mathbf{w}_v^\top\mathbf{x}_i+\mathbf{w}_u^\top\mathbf{x}_i\}\},\end{aligned}\tag{4}$$

where $\phi(\mathbf{w})$ is the loss function of using $\mathbf{w}$ and $\mathbf{x}$ to predict y. We also assume that $\phi(\mathbf{w})$ is convex in $\mathbf{w}$ for

each $\mathbf{x}$ and the other is a simple strongly convex function.

$$\varphi(\mathbf{w}) = \frac{\lambda_{min}}{2}\sum_{v\in\mathbf{Y}}\| \mathbf{w}_v\|^2 = \frac{\lambda_{min}}{2}\| \mathbf{w}\|^2, \quad \textbf{(5)}$$

where $\varphi(\mathbf{w})$ is a l_2 -regularization term. When l_2 -regularization is used with the hinge loss function, we have the standard setup of support vector machines. We assume $\varphi(\mathbf{w})$ is a simple strongly convex function, and its effective domain $\varphi(\mathbf{w}) = \{\mathbf{w}\in\mathbf{R}^n \mid \varphi(\mathbf{w}) < +\infty\}$ is closed. In addition, $\varphi(\mathbf{w})$ is subdifferentiable (a subgradient always exists) on domain $\varphi(\mathbf{w})$.

The hierarchical feature selection algorithm with orthogonal transfer (HiFSOT) is designed in Algorithm 1 to consider the class hierarchical structure. We take measures to selecting features or eliminates redundant features with the weight in the hierarchy. While we make a predict label for test data. We consider multi-class classification problems in which the set of labels are organized hierarchically as a category tree, and the examples are classified recursively from the root to the leaves.

Algorithm 1. Hierarchical Feature Selection with Orthogonal Transfer (HiFSOT)

Input: Examples data $\{(\mathbf{x}_1, y_1),(\mathbf{x}_2, y_2),\cdots,(\mathbf{x}_n, y_n)\}$

1. **for** $v\in\mathbf{Y}$ **do**
2. Compute $\mathbf{w}_v$ according to Eqs. (4) and (5);
3. Sort each feature $f_i|_{i=1}^d$ according to $\mathbf{w}_v$ in descending order;
4. Select the top ranked features of node v;
5. **end for**
6. Return a union of the selected features from all nodes;

We can select feature according to the weight of the class for each node using this algorithm. The features for the parent node is the union of the leaf nodes. The feature subset of the root node is obtained from bottom to top which is listed in Line 6 of Algorithm 1. We can obtain the selected feature subset using the hierarchical class structure.

Because of utilizing the hierarchical structure, the classification problem can be decomposed into a set of smaller problems corresponding to hierarchical splits in the tree. For example, the frequency of the word *sports* is a very indicative feature when we classify movie and sport categories. However, the word *parsing* can be much more indicative than sports when we classify two subclasses basketball and shooting in sports category. In general, classification at different levels of the hierarchy may rely on very different features.

We give an intuitive interpretation to understand the hierarchical feature selection with orthogonal transfer. We utilize protein DD which represents all major structural classes: α, β, α=β and $\alpha + \beta$ in Figure 1. We select the feature from the leaf node to the root node in Figure 1. The internal node has feature subset {438, 425, 435, 432} and {438, 425} that is the union of its child nodes. We use the feature subset of root node as the final feature subset {438, 425, 435, 432} for protein DD dataset.

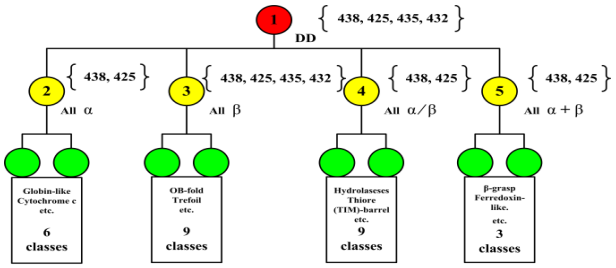

Figure 1. Hierarchical feature selection of DD dataset

3 Experiment

In this section, we test the performance of hierarchical feature selection with orthogonal transfer. To evaluate performance, the algorithm is implemented in MATLAB language and tested on eight datasets. "#" represents the number of selecting features in these tables.

3.1 Datasets

Experiments employ text, protein and image datasets to verify the performance of the proposed algorithm. All these datasets are single labeled and have hierarchical class structures. Some basic statistics about these datasets are given in Table 1.

Table 1. Dataset information

No.	Datasets	|Sample|	|Feature|	|Label|	|Node|	|Leaf|	|Depth|
1	20Newsgroups	3,769	26,214	20	27	20	3
2	DD	3,625	473	27	32	27	3
3	F194	8,525	473	194	202	194	3
4	Bridges	108	11	6	8	6	2
5	Glass	214	9	7	12	7	3
6	SUN	22,556	4,096	324	343	324	3
7	AWAphog	9,607	252	10	17	10	3
8	VOC2010	12,283	1,000	20	30	20	4

Experiments are carried on eight standard datasets obtained from the UCI repository [14]. The text dataset is 20Newsgroups [15]. We also test on two protein datasets including DD [16] and F194 [17]. Five real-world image datasets are Bridges [14], Glass, SUN [18], AWAphog [19], and VOC2010 [20]. AWAphog is formed from Animals with Attributes. VOC2010 is a benchmark in visual object category recognition and detection. The hierarchical tree structure of protein, text, and image datasets are shown in Figure 1, Figure 2, and Figure 3, respectively. The internal nodes and the root node have different colors which are used to distinguish leaf nodes.

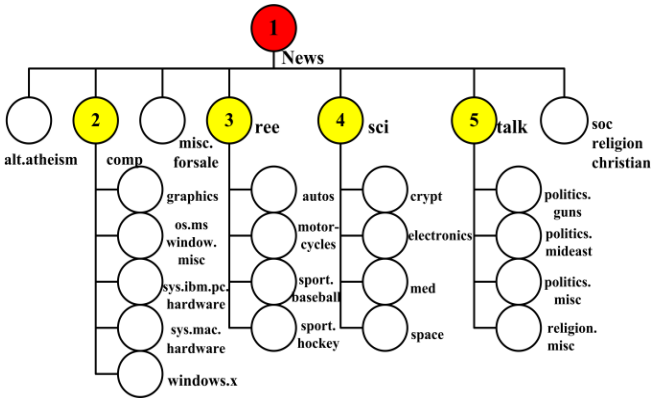

Figure 2. Hierarchical tree structure of 20Newsgroup text dataset

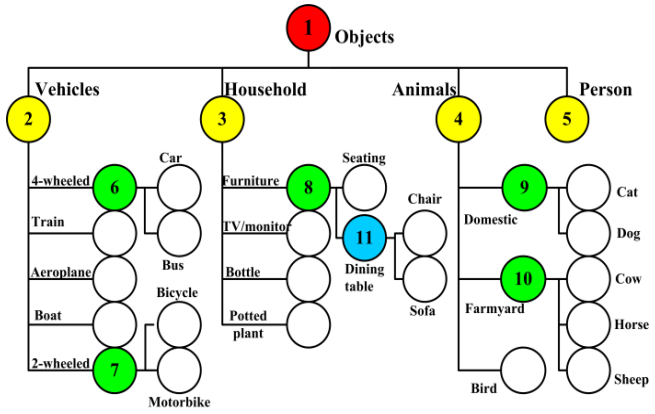

Figure 3. Hierarchical tree structure of image dataset VOC2010

3.2 Comparison Methods

We compare our feature selection method (called as HiFSOT) with several popular feature selection methods using LibSVM [21].

Fisher Score [22]: Fisher score is one of the most widely used supervised feature selection methods. It selects each feature independently according to their scores under the Fisher criterion, which leads to a suboptimal subset of features.

Relief [23]: Relief is a classical feature selection algorithm inspired by instance-based learning.

mRMR: Minimal redundancy maximal-relevance criterion (mRMR) [24] is an effective feature selection scheme which avoids the difficult multivariate density estimation in maximizing dependency.

RFS [25]: Feature selection via joint $l_{2,1}$ -norm minimization which employs joint $l_{2,1}$ -norm minimization on both loss function and regularization to realize feature selection across all data points.

3.3 Parameter Setting

In this section, we consider the parameters of different feature selection algorithms. For Fisher Score, Relief, mRMR, RFS, and HiFSOT, we select same percentage features for all algorithms in Table 2.

Table 2. The number of selected features

No.	Number	Percentage
1	{28,49,69,86,108}	{0.1,0.2,0.3,0.33,0.4}
2	{11,19,27,35,50}	{2,4,6,7,10}
3	{27,32,38,41,44}	{6,7,8,8.7,9}
4	{2,4,6,9}	{18,36,54,81}
5	{2,5,7,8}	{22,55,77,88}
6	{268,500,701,1069,1925}	{6,12,17,26,47}
7	{37,62,103,117,154}	{15,25,40,46,61}
8	{10,30,67,104,164}	{1,3,7,10,15}

As we know, $C \in R$ is the weight adjusting the importance between the regularization term and the loss term. We also tried different values of C varying from 1 to 100 and did not notice any significant difference in classification performance. Thus, we set $C = 1$. We set $\mu = 1$ in Eq. (2) and $\lambda = 1$ in Eq. (5). We employ SVM classifier that individually performed on all datasets using 5-fold cross-validation. We utilize the linear kernel with the parameter $b = 1$. In our experiment, we repeat 5 times and report the average results for each dataset.

3.4 Experimental Results

To evaluate the performance of our hierarchical feature selection algorithms, we illustrate the classification effect from classification performance and statistical perspectives. Demsar advised to statistically compare algorithms on multiple datasets using Friedman test followed by Bonferroni-Dunn test [26]. The best results are enlightened in bold in these tables.

3.5 Results on Text Dataset

Table 3 presents the classification accuracy results on 20Newsgroups dataset. We select feature subset with different number as {28, 49, 69, 86, 108}. We can observe that the accuracy of hierarchy method is better than the flat methods. Only when we select 108 features, the proposed algorithm falls behind Fisher score. The 20Newsgroups dataset is a sparse dataset. The mRMR algorithm is not suitable for this type

dataset.

3.6 Results on Protein Datasets

The comparison results of feature selection algorithms on protein datasets are listed in Tables 4 and 5. The classification results of SVM classifier using different feature selection methods are listed in Table 4. Protein structural class information plays an important role in several aspects. For example, tertiary structure prediction, protein folds prediction, and protein function analysis [27]. It is evidence that our algorithm performs better than other approaches on different number features {11, 19, 27, 35, 50} in most cases. The mRMR feature selection algorithm performs better than other compared approaches.

We select different number features {27, 32, 38, 41, 44} on F194 dataset. It is clear that the proposed method performs better than other approaches on different number features. Moreover, 5-fold cross-validation tests on F194 updated large-scale datasets with varying sequence similarities further confirm that our method is a promising tool for predicting low-similarity protein structural classes.

3.7 Results on Image Datasets

Table 6 and Table 7 show the comparison result using different numbers of selected features on two small datasets Bridges and Glass. On Bridges, the performance of our hierarchy method is better than flat feature selection methods, and it is almost close to the performance with all features when we select nine features. It means that we can obtain the good performance of classification only using representative features. On Glass, the performance of our algorithm is the best result among these methods. The average classification accuracy of our method is 58.46% better than other algorithms.

Table 8, Table 9, and Table 10 show the results on SUN, AWAphog, and VOC2010, respectively. On SUN, the classification results on different features of the HiFSOT algorithm are generally better than that of other flat feature selection algorithms. On AWAphog and VOC2010 datasets, our method converges to a stable value using random initialization different number of selected features. Moreover, the larger number of feature we selected, the less standard deviation of the objective function value.

3.8 Statistical Comparison of Classification Accuracy

The Friedman test reports a significant difference but the post-hoc test fails to detect it. In our experiments the procedure is illustrated by the data from Table 11, which compares four algorithms. It ranks the algorithms for each dataset separately, the best performing algorithm getting the rank of 1, the second best rank 2, and so on. Then, the average ranks of all algorithms on all datasets are calculated and compared.

Table 3. Classification accuracy (%) on 20Newsgroups dataset with different number of features

Algorithm	#28	#49	#69	#86	#108
Fisher	34.23±1.12	43.81±1.32	47.23±1.28	51.74±1.33	**53.81±1.76**
Relief	14.88±1.86	20.88±2.71	22.66±2.88	23.32±2.04	24.36±1.65
RFS	6.630±0.59	7.700±0.47	9.150±0.52	9.840±0.84	11.14±1.05
HiFSOT	**35.90±1.61**	**45.08±0.98**	**49.64±1.08**	**51.79±1.71**	53.78±1.41

Table 4. Classification accuracy (%) on DD dataset with different number of features

Algorithm	#11	#19	#27	#35	#50
Fisher	39.61±1.15	47.87±1.57	49.30±2.44	53.27±2.71	58.07±1.99
Relief	48.11±1.25	53.76±1.95	63.28±1.20	65.60±1.22	77.49±2.02
mRMR	**56.72±2.01**	**68.16±1.71**	74.37±1.22	76.77±1.74	79.56±0.80
RFS	47.94±1.59	65.90±1.20	74.12±2.29	76.41±1.54	79.78±1.46
HiFSOT	53.71±1.63	65.13±2.32	**74.96±1.32**	**77.13±1.12**	**79.92±1.35**

Table 5. Classification accuracy (%) on F194 dataset with different number of features

Algorithm	#27	#32	#38	#41	#44
Fisher	32.99±1.62	33.49±1.52	36.39±1.90	40.25±2.30	42.51±2.77
Relief	46.44±1.91	51.46±1.09	57.31±0.81	59.45±0.83	61.54±0.73
mRMR	57.07±1.21	59.48±1.21	60.75±1.33	61.35±1.24	61.99±1.01
RFS	46.98±0.68	54.15±1.37	58.77±1.15	60.36±1.32	61.71±0.83
HiFSOT	**57.78±0.92**	**60.12±1.01**	**61.56±1.37**	**61.86±1.34**	**62.59±1.24**

Table 6. Classification accuracy (%) on Bridges dataset with different number of features

Algorithm	#2	#4	#6	#9	Average
Fisher	51.69±10.0	59.05±11.1	61.77±13.4	63.68±11.5	59.05±11.5
Relief	44.54±11.7	54.67±9.75	62.94±9.36	61.04±9.97	55.80±10.2
mRMR	40.65±5.15	48.96±9.33	49.00±9.61	60.26±7.82	49.72±7.98
RFS	54.50±10.9	**59.18±12.4**	62.90±11.6	58.22±7.32	58.70±10.5
HiFSOT	**55.63±6.29**	56.62±9.39	**63.07±7.97**	**64.89±11.1**	**60.05±8.69**

Table 7. Classification accuracy (%) on Glass dataset with different number of features

Algorithm	#2	#5	#7	#8	Average
Fisher	40.22±11.2	58.89±8.39	58.89±6.16	59.37±6.53	54.34±8.08
Relief	**49.99±5.93**	54.68±7.72	56.57±7.60	57.51±7.77	54.69±29.0
mRMR	48.13±4.16	58.90±4.56	59.83±3.55	59.83±3.55	56.64±15.8
RFS	43.95±8.48	59.32±6.24	61.65±6.34	60.72±6.26	56.41±6.83
HiFSOT	44.40±8.56	**64.01±6.95**	**64.92±6.23**	**64.52±4.80**	**58.46±6.63**

Table 8. Classification accuracy (%) on SUN dataset with different number of features

Algorithm	#268	#500	#701	#1069	#1925
Fisher	58.52±0.45	64.68±0.42	66.31±0.45	67.61±0.37	68.54±0.52
Relief	59.60±1.33	64.21±0.62	66.07±0.64	67.56±0.62	68.31±0.72
mRMR	61.83±0.44	65.07±0.38	66.27±0.41	67.51±0.80	68.31±0.45
HiFSOT	**62.39±0.80**	**65.52±0.30**	**66.95±0.61**	**67.79±0.77**	**68.63±0.63**

Table 9. Classification accuracy (%) on AWAphog dataset with different number of features

Algorithm	#37	#62	#103	#117	#154
Fisher	23.95±1.28	27.36±0.96	30.23±0.83	30.76±1.09	31.10±1.39
Relief	23.61±1.38	26.53±1.08	29.42±1.30	30.03±1.18	30.51±0.93
mRMR	23.54±0.98	26.20±1.24	28.91±1.03	29.43±1.16	30.73±0.55
RFS	24.93±1.10	**28.42±0.91**	30.01±0.70	30.53±0.72	31.34±0.92
HiFSOT	**26.10±0.68**	28.33±1.37	**30.26±1.37**	**31.03±0.71**	**31.58±1.46**

Table 10. Classification accuracy (%) on VOC2010 dataset with different number of features

Algorithm	#10	#30	#67	#104	#164
Fisher	28.80±0.34	29.15±0.40	32.30±0.44	33.44±0.36	34.29±0.24
Relief	28.19±0.63	28.80±0.58	31.24±0.92	33.55±0.75	35.39±0.65
mRMR	28.71±0.94	30.17±1.04	33.26±1.16	33.85±0.85	36.71±1.05
RFS	28.22±0.62	29.11±0.52	**32.60±0.71**	33.70±0.38	**36.90±1.00**
HiFSOT	**28.93±0.70**	**30.54±0.80**	32.56±1.49	**34.20±0.79**	36.28±0.60

Table 11. Comparison of classification accuracy between various feature selection and several datasets

No.	Datasets	Fisher	Relief	mRMR	RFS	HiFSOT
1	20Newsgroups	46.16(2)	21.22(3)	---(5)	8.890(4)	**47.24(1)**
2	DD	49.62(5)	61.65(4)	**71.12(1)**	68.83(3)	70.17(2)
3	F194	37.19(5)	55.24(4)	60.13(2)	56.39(3)	**60.78(1)**
4	Bridges	59.05(2)	55.80(4)	49.72(5)	58.70(3)	**60.05(1)**
5	Glass	53.34(5)	54.69(4)	56.64(2)	56.41(3)	**58.46(1)**
6	SUN	65.13(4)	65.15(3)	65.79(2)	---(5)	**66.26(1)**
7	AWAphog	28.88(3)	28.02(4)	27.76(5)	29.04(2)	**29.46(1)**
8	VOC2010	31.60(4)	31.43(5)	**32.54(1)**	32.11(3)	32.50(2)
9	Avg.rank	3.75	3.875	2.875	3.25	1.25

The number of parentheses represents the average classification performance. The last line is average rank in the table. Average ranks by themselves provide a fair comparison of the algorithms. On average, mRMR and RFS rank the second and third (with ranks 2.875 and 3.25, respectively), and Fisher and Relief rank the four and five (3.75 and 3.875). As shown in Table 11. The Friedman test checks whether the measured average ranks are significantly different from the mean rank $R_j = 2.5$ expected under the null-

hypothesis: $\chi_F^2 = 14.3$ and $F_F = 7.59$.

We find $\alpha = 0.005$ according to $\chi_F^2 = 14.3$ from any statistical book. With five algorithms and eight datasets, F_F is distributed according to the F distribution with 4 and 28 degrees of freedom. The critical value of $F(4, 28)$ for $\alpha = 0.005$ is 4.74. The test result is $p = 0$ from standard normal distribution. All the five feature selection algorithms are different in terms of proportion of selected features. So we reject the null-hypothesis.

In order to further explore feature selection algorithms whose reduction rates have statistically significant differences, we performed a Nemenyi test [28]. Considering that it belongs to a statistical nonsense since a subject cannot come from two different populations. The other possible hypothesis made before collecting the data could be that it is possible to improve on HiFSOT performance by tuning its parameters. The easiest way to verify this is to compute the *CD* with the test, and *CD* = 1.80. The ranks in the parentheses are used in computation of the Friedman test.

We connect the groups of algorithms that are not significantly different in Figure 4. We can mark the interval of one *CD* to the left and right of the average rank of the control algorithm. The results indicate that the hierarchical HiFSOT is statistically better than those of Fisher, Relief, mRMR and RFS. There is no consistent evidence to indicate statistical classification accuracy different among HiFSOT and mRMR.

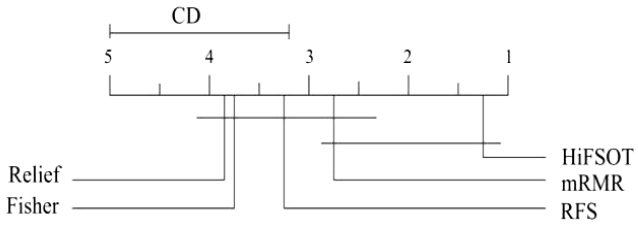

Figure 4. Comparison of classification accuracy for various feature selection algorithms against each other with the Nemenyi test

4 Conclusions and Future Work

Although many real-world classification systems have complex hierarchical structures, few learning methods capitalize on this structure. In this paper, we have proposed a hierarchical feature selection algorithm with orthogonal transfer method. It has decreased the size of the structure without significantly decreasing prediction precision of the classifier built using only the selected features. Experimental results indicate the efficiency of the proposed algorithm. With regard to future research, much work needs to be undertaken. First, the current implementation of the algorithm deals only with the tree class structure. Secondly, the current implementation of the algorithm deals only with the weight of the feature to the category problems that is the principal limitation. In the future, the extending algorithm needs to be proposed to cope with DAG class structure. The extending algorithm needs to be proposed to cope with multivariate class problems. In summary, this study suggests new research trends concerning hierarchical classification, feature selection problem and relative dependency learning.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 61703196 and the Natural Science Foundation of Fujian Province under Grant No. 2018J01549.

References

[1] X. D. Wu, X. Q. Zhu, G. Q. Wu, W. Ding, Data Mining with Big Data, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 1, pp. 97-107, January, 2014.

[2] Q. Hu, L. Zhang, Y. Zhou, W. Pedrycz, Large-scale Multi-modality Attribute Reduction with Multi-kernel Fuzzy Rough Sets, *IEEE Transactions on Fuzzy Systems*, Vol. 26, No. 1, pp. 226-238, February, 2018.

[3] J. Dai, Q. Hu, J. Zhang, H. Hu, N. Zheng, Attribute Selection for Partially Labeled Categorical Data by Rough Set Approach, *IEEE Transactions on Cybernetics*, Vol. 47, No. 9, pp. 2460-2471, September, 2017.

[4] G. Tong, L. Mei, N. Xiong, An Improved Filtering Method Applied in Digital Mobile Terminal for Images Getting from Wireless Network Remote Transmission, *Journal of Internet Technology*, Vol. 19, No. 3, pp. 641-648, January, 2018.

[5] S. M. Villela, S. D. C. Leite, R. F. Neto, Feature Selection from Microarray Data via an Ordered Search with Projected Margin, *International Conference on Artificial Intelligence, AAAI Press*, Buenos Aires, Argentina, 2015, pp. 3874-3881.

[6] P. Ristoski, H. Paulheim, Feature Selection in Hierarchical Feature Spaces Discovery Science, *Springer International Publishing*, 2014.

[7] A. Goswami, A. K. Das, V. Odelu, A Secure and Efficient Time-bound Hierarchical Access Control Scheme for Secure Broadcasting, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 22, No. 4, pp. 236-248, January, 2016.

[8] S. Dumais, H. Chen, Hierarchical Classification of Web Content, *International ACM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, 2000, pp. 256-263.

[9] S. Bengio, J. Weston, D. Grangier, Label Embedding Trees for Large Multi-class Tasks, *International Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2010, pp. 163-171.

[10] Cesa-Bianchi, Nicolò, D. U. Itgentile, Claudio, U. Itzaniboni,

Incremental Algorithms for Hierarchical Classification, *Journal of Machine Learning Research*, Vol. 7, No. 1, pp. 31-54, December, 2006.

[11] B. E. Burke, J. L. Tonry, M. J. Cooper, D. J. Young, A. H. Loomis, P. M. Onaka, G. A. Luppino, Development of the Orthogonal-transfer Array, in: M. M. Blouke (Ed.), *Proceeding of the SPIE*, Vol. 6068, SPIE, 2006, pp. 173-180.

[12] D. Zhou, L. Xiao, M. Wu, Hierarchical Classification via Orthogonal Transfer, *International Conference on International Conference on Machine Learning*, Bellevue, WA, 2011, pp. 801-808.

[13] X. Lin, Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization, *Journal of Machine Learning Research*, Vol. 11, No. 1, pp. 2543-2596, March, 2010.

[14] C. Blake, C. J. Merz, *UCI Repository of Machine Learning Databases*, University of California, 1998.

[15] K. Lang, Newsweeder: Learning to Filter Netnews, *12th Machine Learning Proceedings*, Tahoe, CA, 1995, pp. 331-339.

[16] C. H. Ding, I. Dubchak, Multi-class Protein Fold Recognition Using Support Vector Machines and Neural Networks, *Bioinformatics*, Vol. 17, No. 4, pp. 349-358, April, 2001.

[17] L. Wei, M. Liao, G. Xing, Q. Zou, An Improved Protein Structural Classes Prediction Method by Incorporating both Sequence and Structure Information, *IEEE Transactions on Nanobioscience*, Vol. 14, No.4, pp. 339-349, June, 2015.

[18] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, A. Oliva, SUN Database: Exploring a Large Collection of Scene Categories, *International Journal of Computer Vision*, Vol. 119, No. 1, pp. 3-22, August, 2016.

[19] C. H. Lampert, H. Nickisch, S. Harmeling, Learning to Detect Unseen Object Classes by Between-class Attribute Transfer, *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, 2009.

[20] M. Everingham, J. Winn, The PASCAL Visual Object Classes Challenge: A Retrospective, *Int. J. Comput. Vis.*, Vol. 111, pp. 98-136, 2015.

[21] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, 1981.

[22] V. B. Zubek, T. G. Dietterich, Pruning Improves Heuristic Search for Cost-Sensitive Learning, *Nineteenth International Conference on Machine Learning*, The University of New South Wales, Sydney, Australia, 2002, pp. 19-26.

[23] K. Kira, L. A. Rendell, A Practical Approach to Feature Selection, *International Workshop on Machine Learning*, Aberdeen, Scotland, United Kingdom, 1992, pp. 249-256.

[24] H. Peng, F. Long, C. Ding, Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 27, No. 8, pp. 1226-1238, August, 2005.

[25] F. Nie, H. Huang, X. Cai, C. Ding, Efficient and Robust Feature Selection via Joint $l_{2,1}$ -norms Minimization, *International Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2010, pp. 813-1821.

[26] J. Ar, Statistical Comparisons of Classifiers over Multiple Data Sets, *Journal of Machine Learning Research*, Vol. 7, No. 1, pp. 1-30, December, 2006.

[27] K. C. Chou, Prediction of Protein Structural Classes and Subcellular Locations, *Current Protein & Peptide Science*, Vol. 1, No. 2, pp. 171-208, September, 2000.

[28] P. Nemenyi, *Distribution-free Multiple Comparisons*, Ph.D. Thesis, Biometrics, United States, 1963.

Biographies

Limei Dong received a master degree from Minnan Normal University in 2018. Her research interest is data mining.

Hong Zhao received the Ph.D degree from Tianjin University, Tianjin, China, in 2019. She is currently a Professor of the School of Computer Science and the Fujian Key Laboratory of Granular Computing and Application, Minnan Normal University, Zhangzhou, China. Her current research interests include rough sets, granular computing, and data mining for hierarchical classification.