



Cost-sensitive hierarchical classification via multi-scale information entropy for data with an imbalanced distribution

Weijie Zheng¹ · Hong Zhao^{1,2}

Accepted: 18 November 2020 / Published online: 20 January 2021
© Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Imbalanced distributions present a great problem in machine learning classification tasks. Various algorithms based on cost-sensitive learning have been developed to address the imbalanced distribution problem. However, classes with a hierarchical tree structure create a new challenge for cost-sensitive learning. In this paper, we propose a cost-sensitive hierarchical classification method based on multi-scale information entropy. We construct an information entropy threshold for each level in the tree structure and assign cost-sensitive weights accordingly. First, we use the class hierarchy to divide a large hierarchical classification problem into several smaller sub-classification problems. In this way, a large-scale classification task can be decomposed into multiple, controllable, small-scale classification tasks. Second, we use a logistic regression algorithm to obtain the probabilities of classes at each level. Then, we consider the information entropy at each level as a threshold, which decreases inter-level error propagation in the tree structure. Finally, we design a cost-sensitive model based on the information of each class and use hierarchical information entropy weights as cost-sensitive weights. Information entropy measures the information of the majority and minority classes and allocates them different cost weights to solve imbalanced distribution problems. Experiments on four imbalanced distribution datasets demonstrate that the cost-sensitive hierarchical classification algorithm provides excellent efficiency and effectiveness.

Keywords Imbalanced distribution · Information entropy · Cost-sensitive · Hierarchical classification

1 Introduction

The imbalanced distribution problem is a special classification task in machine learning [22, 36, 44] in which the minority classes include a smaller number of samples than the majority classes. Therefore, there is a large number of minority classes, which are ignored and difficult to model effectively. The imbalanced distribution problem exists in numerous areas [7, 12, 29], such as image classification [11], age estimation [24], cancer diagnosis [28], and face recognition [47].

There are three challenges in addressing the imbalanced distribution problem. First of all, the minority classes are

an important part of the whole dataset. Thus, we should consider them as seriously as the majority classes. Secondly, traditional classifiers were initially designed for balanced datasets. The lack of ability leads to hyperplane bias in the majority classes. Thirdly, the samples in the minority classes cannot contain rich information. The samples in different classes cannot be correctly recognized.

The existing approaches above have tried to tackle the imbalanced problem in binary and multi-classification. However, many datasets contain dozens or hundreds of imbalanced distribution classes, which are organized in hierarchical trees. Hierarchical tree structures exist in various applications, such as image classification [17, 50], protein classification, and texture classification [35]. The increasing number of classes within a hierarchical structure [48] provides great supplementary information for solving imbalanced distribution problems.

In this paper, we combine hierarchical classification [10, 26, 38] and cost-sensitive learning [25] to address imbalanced distribution datasets. The algorithm takes advantage of the hierarchical information of classes in datasets to design a cost-sensitive model. First, we make use of the

✉ Hong Zhao
hongzhaocn@163.com

¹ Fujian Key Laboratory of Granular Computing and Application, Minnan Normal University, Zhangzhou, Fujian, 363000, China

² School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, 363000, China

hierarchical structure to divide a large-scale classification problem into several small-scale ones. Small-scale classification tasks are much easier to control and compute. Second, the logistic regression algorithm is used to calculate the probabilities of all classes on each level. This model obtains more accurate classification probabilities without assuming the distribution of the datasets. The hierarchical structure provides hierarchical information of different classes. Then, we set a threshold based on information entropy to determine the cost-sensitive factor for balancing the majority and minority classes. The threshold extracts and measures the information of the majority and minority classes to decide the cost-sensitive factor that suits each class. The threshold increases the accuracy of minority classes and reduces inter-level error propagation on each level.

Finally, we design a cost-sensitive model based on hierarchical information entropy [6] and the information [39, 41] of different classes. Two cost-sensitive methods provide different punishments to prevent the classifier from biasing the majority classes. We compare the proposed model with the traditional hierarchical classification approach. In terms of improving the overall accuracy, the accuracies of the minority class are increased significantly.

The contributions of this paper are as follows:

- We build a quick, cost-sensitive, hierarchical classification model to address imbalanced distribution data problems. It is different from existing cost-sensitive classification models, mainly on hierarchical structure.
- We construct an adaptive threshold strategy instead of a fixed threshold value. With this strategy, the threshold value is decided by the data rather than given by users.
- We compare the proposed algorithm with seven state-of-the-art hierarchical algorithms on five imbalanced distribution datasets to demonstrate its great efficiency and effectiveness.

The remainder of this paper is presented as follows. In Section 3, we introduce the proposed cost-sensitive hierarchical classification algorithm. The experiment settings about datasets, compared algorithms and evaluation metrics are given in Section 4. Experimental results and discussion about hierarchical classification on imbalanced distribution datasets are in Section 5. Finally, we present the conclusions and future work of this paper in Section 6.

2 Related work

Resampling [14, 23, 32] and cost-sensitive learning [13] are two traditional solutions to the imbalanced distribution problem.

The resampling method solves this problem by balancing the numbers of samples in the majority and minority classes.

The resampling method changes the original data structure, which loses precious data. The resampling methods include undersampling of the majority classes and oversampling of the minority classes. Liu et al. [20] used two undersampling strategies which include the cluster centers to represent the majority class and the nearest neighbors of the cluster centers to address the class imbalanced data. Castellanos et al. [5] proposed an approach based on adapting the well-known synthetic minority oversampling technique algorithm to the string space.

Cost-sensitive learning is a powerful tool for solving the imbalanced distribution problem without altering the original data. Researchers have done much work to address imbalanced distribution problems with cost-sensitive methods [21, 37, 46]. For instance, Sahin et al. [30] proposed a cost-sensitive decision tree algorithm for minimizing the sum of misclassification costs and improving the true-positive proportion of datasets. Similarly, Braytee et al. [2] addressed a cost-sensitive strategy using the imbalanced proportion of classes for feature extraction to punish the majority classes. In addition, Cao et al. [4] designed a novel framework for training a cost-sensitive classifier that is driven by imbalanced evaluation criteria.

Some researchers have focused on cost-sensitive practical applications [33]. For instance, Sheng et al. [34] constructed a decision tree to decrease the total misclassification costs and feature costs of medical diagnostic tests. Zhang et al. [47] proposed two cost-sensitive methods using Bayes decision theory and a k -NN classifier for solving group face-recognition tasks.

3 Cost-sensitive hierarchical classification algorithm

3.1 Basic framework

The framework involves cost-sensitive hierarchical classification via multi-scale information entropy for data with an imbalanced distribution. It contains two cost-sensitive weighting processes and an important threshold strategy. The proposed framework is listed in Fig. 1, which contains the following three parts:

- (1) The first part of Fig. 1 introduces the hierarchical structure used for imbalanced data and hierarchical classification. The hierarchical class structure and number of samples are listed in this tree. For instance, there are three sub-classes of Class *Animal*, which number 500, 30, and 300, respectively. The number of samples varies greatly from class to class, which results in imbalanced data with a hierarchical structure. For hierarchical classification, the classifier classifies

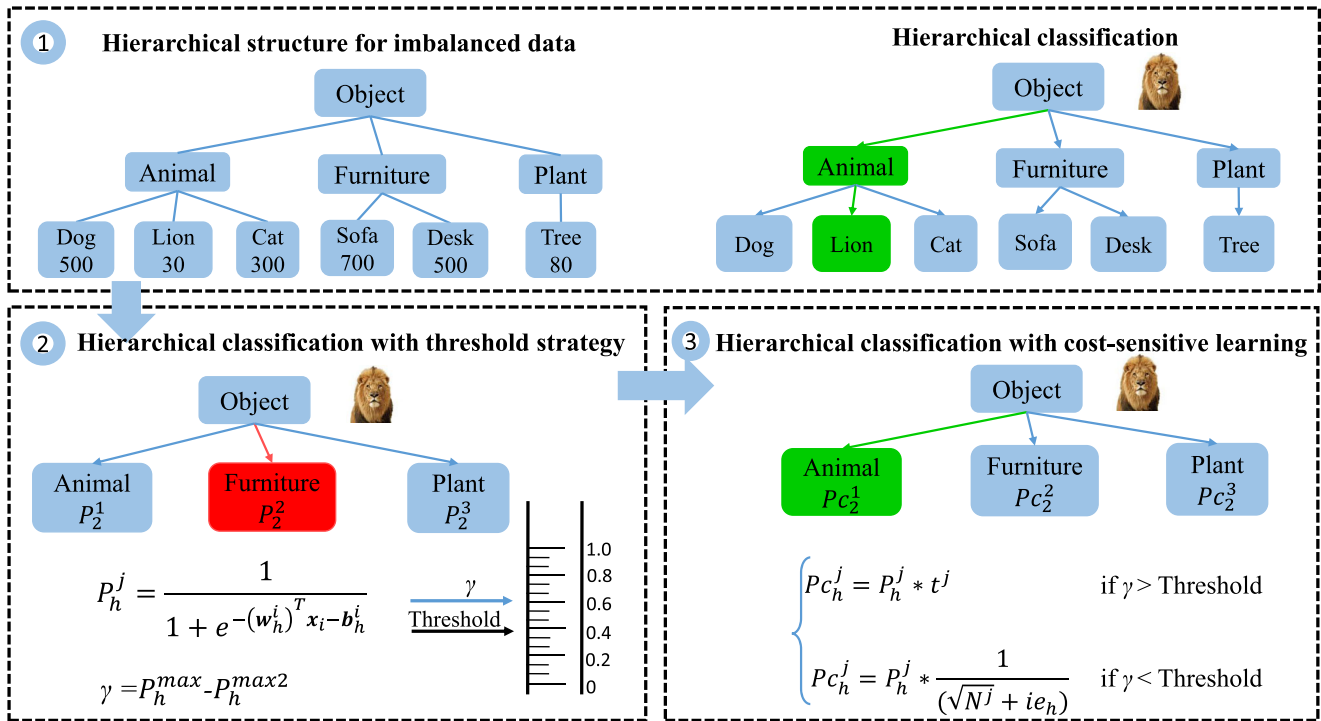


Fig. 1 The framework of cost -sensitive hierarchical classification via multi-scale information entropy for data with an imbalanced distribution

the samples from coarse- to fine-grained. For the new samples, we first divide them into coarse classes and then subdivide them into fine classes.

- The second part of Fig. 1 presents hierarchical classification with a threshold strategy. The classifier computes the probability on the second level. The algorithm computes the difference between the two maximum probabilities. The algorithm computes information entropy, based on the probability of each class, as the threshold value.
- The third part of Fig. 1 introduces hierarchical classification with cost-sensitive learning. The algorithm gives different cost-sensitive weights to different classes based on the threshold. The classifier selects the class according to the weighted probability.

- Root node is Node *Object*;
- Node *Animal* is a sibling node of Node *Plant*;
- Node *Animal* has three leaf nodes.

Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ be a data matrix, where m is the number of samples, and n is the number of features. We use \mathbf{x}_i to represent the i -th sample, $\mathbf{x}_i \in \mathbb{R}^{m \times n}$ and $\mathbf{X} = \{\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_m\}$. Let $\mathbf{Y} \in \mathbb{R}^{m \times 1}$ be a class vector. We use y_i to represent the class of \mathbf{x}_i , where $y_i \in \mathbf{Y}$ and $\mathbf{Y} = \{y_1; y_2; \dots; y_m\}$.

Logistic regression is a great classification algorithm used in supervised learning and has many advantages. It can directly model the classification possibility without assuming the data distribution in advance, thus avoiding

3.2 Hierarchical structure and classification

We decompose a large-scale classification task into multiple controllable small-scale classification tasks. All the classes are grouped into the hierarchical tree structure. The root node is at the top of the hierarchical tree structure. A sibling node means that two or more nodes share the same parent node. Nodes on the second level have sibling and child nodes. Leaf nodes only have sibling nodes. An example of a hierarchical tree structure is shown in Fig. 2. From this figure, we have the following:

Hierarchical classification

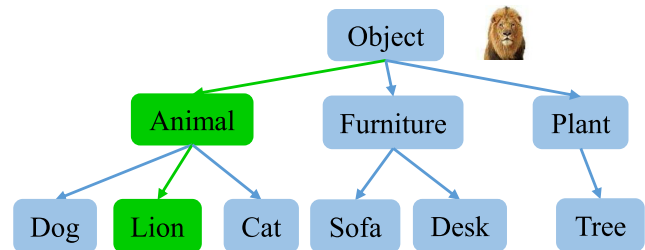


Fig. 2 Hierarchical classification tree structure ($h = 3$)

problems caused by an inaccurate hypothesis distribution. It can not only predict the category but can also predict the approximate probability. The computing cost is low and large amounts of data can be processed with few resources. We can obtain parameter w and parameter b after training the training set of the dataset. Parameter w is a matrix and parameter b is a vector. We use a sigmoid function for the probability p . The sigmoid function is denoted as follows:

$$p = \frac{1}{1 + e^{-w^T x_i - b}} \tag{1}$$

In the hierarchical tree structure, logistic regression first computes the probability of every node from the root node to the leaf node. The probability that the i -th sample belongs to j -th class on the h -th level is computed as the following equation:

$$p_h^j = \frac{1}{1 + e^{-(w_h^j)^T x_i - b_h^j}} \tag{2}$$

We define parameter p_h^{max} as the maximum probability from all p_h^j on the h -th level. Maximum probability means that the node has the shortest distance between itself and the hyperplane that the classifier builds. The classifier selects that node after comparing it with the probability of nodes on the same level. The classifier computes the probability of the child nodes of that node and selects the child node with the maximum probability. In a top-down process, the classifier repeats these two steps until the end is reached. We can understand the process of hierarchical classification better by referring to Fig. 2. The classifier determines the probabilities of Nodes *Animal*, *Furniture* and *Plant*, which are 0.2, 0.6, and 0.2, respectively. The classifier selects the Node *Furniture* as the best choice. The classifier makes the right choice on this level and computes the probability of nodes that belong to Node *Furniture* on the next level. The classifier still selects the node with the maximum probability on this level. The leaf label finally selected is the predicted label round after round.

3.3 Hierarchical classification with threshold

Hierarchical classification simply and directly selects the node with the maximum probability as the best choice on one level. However, there is only a slight gap between the maximum probability and others. Direct selection of the classifier leads to errors. The errors are passed to other branches along the tree structure, such that the classifier cannot obtain the correct classification and classification accuracy is reduced. To avoid the classifier directly selecting that node straight, the threshold is used as a judgment

condition. We set the threshold for re-measuring the node with maximum probability.

The datasets we build are a collection of information. We extract and measure the information we want from all aspects of the classification. We make decisions about what to do based on mathematical expectations in real life, and information entropy is the mathematical expectation of a random variable. We use information entropy as a quantitative index of the information content of a system so that it can be further used as a criterion for the parameter selection. In the information entropy formula, the negative logarithm of probability indicates the amount of information carried by possible events. The sum of all possible information multiplied by the probability of its occurrence represents the expected value of all information in the whole system.

We define parameter ie as the value of the information entropy factor on each level. Parameter ie can be computed as per the following equation:

$$ie = \frac{1}{2} * \left(1 - \sum_{j=1}^k p_h^j \log_2 p_h^j \right), \tag{3}$$

where j is the j -th class on this level, and k is the number of classes on this level. We add one to the information entropy value and multiply it by one in two. This step is taken to prevent the information entropy value being >1 . We know that the difference between the two probabilities is <1 . We define parameter p_h^{max2} as the second maximum probabilities on the h -th level, and parameter γ as the difference between p_h^{max} and p_h^{max2} . We compare γ with ie and assign a cost weight.

3.4 Hierarchical classification with cost-sensitive learning

After judgment of the threshold, the cost-sensitive method is used. We design a cost-sensitive model to re-measure the importance of the node with maximum probability if γ is greater than the value of the threshold. It combines the prior and posterior probabilities, which avoids the subjective bias of using only the prior probability and the overfitting phenomenon of using the sample information alone. Information entropy is composed of the uncertainty of each node, and the variation in uncertainty between nodes causes them to contribute different proportions to the information entropy. In this way, we set the proportion of information entropy as the posterior probability to determine the greatest importance of each node.

In the hierarchical tree structure, the root node at the top contains the coarse-grained information. By contrast, the leaf node at the bottom has fine-grained information.

Different levels of precision result in different information weights, and different levels have different information weights. An imbalanced distribution dataset is built with classes containing more or fewer samples than others. The more similar the numbers of samples in each class, the more fair the classifier is. We consider hierarchical information entropy and the number of samples in each class as two cost-sensitive factors for balancing the gap between the majority classes and minority classes. The value of hierarchical information entropy at the h level can be denoted as ie_h and can be computed with the following equation:

$$ie_h = \frac{1}{2} * L_h * \left(1 - \sum_{j=1}^k p_h^j \log_2 p_h^j \right), \tag{4}$$

where L_h is the weight of the h -th level in the hierarchical tree structure, h is the height of the hierarchical tree. We set the level of the root node as L_0 . Since L_0 is the root node of the hierarchical tree, the weight of level L_0 is set to 0. We set the weight to be the same as the number of levels. For example, the weight of level L_1 is set to 1. We can understand the process of calculating hierarchical information entropy on each level more easily by referring to Fig. 3.

The differences between the majority and minority classes include the numbers of samples and the proportions of each child class. The majority classes have more child classes than the minority classes. We define parameter t^j as the proportion of child class of the i -th class. Similarly, we count the number of samples of the j -th class as parameter N^j . After comparing the threshold, we propose two cost weights to address the imbalanced distribution of datasets. We multiply p_h^j by t^j to solve the imbalanced problem of child classes on the upper level if γ is greater than ie . We offer a cost weight if γ is less than ie . We consider ie_h and N^j as a cost weight for balancing the sample gap between the majority and minority classes. We take the square root of N^j as a factor to avoid an extremely imbalanced distribution of sample numbers. We define parameter pc_h^j

as the probability of the j -th class on the h -th level with a cost-sensitive weight. We compute pc_h^j using the following equation:

$$pc_h^j = p_h^j * \frac{1}{(\sqrt{N^j} + ie_h)}. \tag{5}$$

In summary, we use information entropy as the threshold in the hierarchical classification system. The cost-sensitive model will be used for measurement work if γ is greater than the threshold. The cost-sensitive model is based on the posterior probability and the proportion of classes. This process re-measures and confirms the advantage of the node with the maximum probability on this level. We use hierarchical information entropy and the number of samples in each class as a cost-sensitive weight. We find the square roots of these numbers to balance and reduce the gap between the majority and minority classes. The cost-sensitive algorithm supports the minority classes for higher costs and reduces the gap between the high and low levels if γ is less than the threshold.

An example of a cost-sensitive hierarchical algorithm with a threshold is shown in Fig. 4. We make $p_1^2 = 0.1$, $p_1^3 = 0.8$ and $p_1^4 = 0.1$. The value of ie is equal to 0.95, so the value of γ is less than the threshold. After reassigning weights, we obtain Class 3 as the best choice. We obtain $p_2^5 = 0.1$ and $p_2^6 = 0.9$. The value of ie is approximately equal to 0.71 and the value of γ is greater than the threshold. The cost-sensitive hierarchical algorithm assigns different cost weights to different classes based on two conditions. Finally, we obtain the best leaf node as Class 5.

The process of cost-sensitive hierarchical classification for imbalanced distribution datasets based on information entropy (CSHC) is set out in Algorithm 1. The threshold strategy of CSHC is introduced in Line 11 of Algorithm 1, and two cost-sensitive weights of CSHC are illustrated in Lines 12, 13, 15, and 16 in Algorithm 1. The computational complexity of the CSHC algorithm is $O(cmn)$, where c is the number of classes, m represents the number of samples, and n is the number of features.

Hierarchical Information Entropy

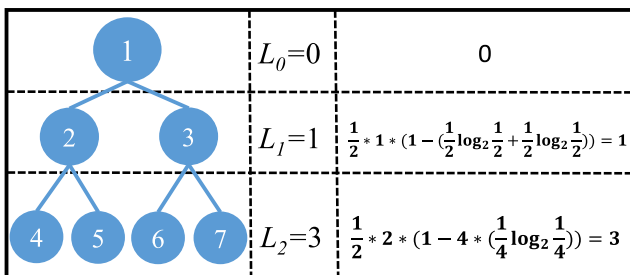


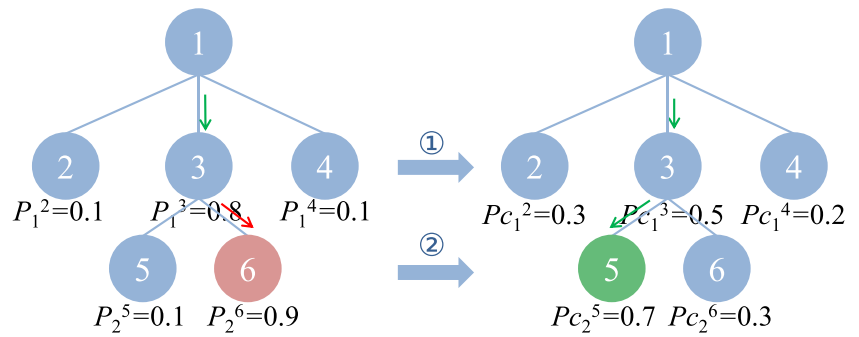
Fig. 3 The calculation process of hierarchical information entropy on each level

4 Experimental settings

We used 10-fold cross-validation in these experiments. We used a computer with an Intel Core i7-3770 processor, 16 GB of memory, and the Windows 10 operating system.¹ The five imbalanced distribution datasets in our experiments are introduced in Section 4.1. Seven state-of-the-art algorithms used for comparison with our algorithm are introduced in

¹Datasets and code used in this research have been uploaded to GitHub. They are accessible at: <https://github.com/fhqxa/CSHC>.

Fig. 4 Comparison of hierarchical classification and cost-sensitive hierarchical classification with the threshold



Section 4.2. In Section 4.3, we introduce the two evaluation metrics used in our experiments.

Algorithm 1 Cost -sensitive hierarchical classification via multi-scale information entropy for data with an imbalanced distribution.

Input: Sample $\mathbf{x}_i = \{\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_m\}$, class $y_i = \{y_1; y_2; \dots; y_m\}$, and h is the height of tree structure.

Output: L is the predicted label.

```

1: for  $e = 1 : h$  do
2:   Initialize parameters  $w_h, b_h$ 
3:   For epoch in epochs:
4:     Caculate loss,  $dw_h, db_h$ 
5:     Update  $w_h$  with  $dw_h$ 
6:     Update  $b_h$  with  $db_h$ 
7:     Update  $w_h, b_h$ 
8:     Update  $p_h^i$ ;
9:     Compute  $ie$  and  $t^i$  on this level;
10:    Update  $p_h^{max}$  and  $p_h^{max2}$  using  $p_h^i$ ;
11:    if ( $\gamma > ie$ ) then
12:       $pc_h^i = p_h^i * (t^i)$ ;
13:      Select  $i$ -th class according to  $pc_h^{max}$ ;
14:    else
15:       $pc_h^i = p_h^i * 1/(\sqrt{N^i} + ie_h)$ ;
16:      Select  $i$ -th class according to  $pc_h^{max}$ ;
17:    end if
18:  end for
19: Return  $L$ ;

```

4.1 Datasets

A description of five experimental datasets is given in Table 1. The description includes the number of sample, feature, class, and height in these datasets.

ILSVRC57 [19]: The dataset is a subset of a public dataset named WordNet. All classes are organized as same as the hierarchical structure in WordNet dataset. The data of ILSVRC57 are built in a three-level tree with 65 categories. There are 57 leaf nodes and 4,096 features. These 57 leaf nodes are basically divided into 5 classes (Bird, Cat,

Dog, Boat, and Car). The leaf node with the most samples involves 221 samples, and the leaf node with the least samples includes 155 samples.

CAR196: CAR196 is an image classification dataset which has 196 types of car pictures. There are 206 nodes built in a three-level tree. It contains 15,685 samples into 196 leaf nodes. The leaf node with the most samples contains 67 samples, and the leaf node with the least samples has 18 samples.

SUN [43]: SUN dataset is modified by a scene understanding image classification dataset. There are 343 nodes organized in a four-level height tree. 22,556 samples are divided into 324 leaf nodes. The leaf node with the most samples contains 1,075 samples, and the leaf node with the least samples includes 36 samples.

DD [8]: DD is a protein dataset that contains protein sequences. Protein sequences are represented by 473 features and divided into 27 important classes. It has a three-level hierarchical tree structure that contains 4 non-leaf nodes on the second level and 27 leaf nodes on the third level. The leaf node with the most samples contains 361 samples, and the leaf node with the least samples contains 17 samples.

F194 [42]: F194 is also a protein dataset with a three-level height tree. It includes 7 non-leaf nodes on the second level and 194 leaf nodes on the third level. These 194 leaf nodes stand for 194 protein classes. In these 194 classes, there are 361 samples in the leaf node with the most samples, but only 10 samples in the leaf node with the least samples.

Table 1 Data description

No.	Dataset	Sample	Feature	Node	Leaf	Height
1	ILSVRC57	11,845	4,096	65	57	4
2	CAR196	15,685	4,096	206	196	3
3	SUN	22,556	4,096	343	324	4
4	DD	3,625	473	32	27	3
5	F194	8,525	473	202	194	3

4.2 Compared methods

- (1) HFSNM: FSNM is a feature selection [3, 40] method based on $l_{2,1}$ -norms minimization [27]. FSNM based on loss function makes the process of feature selection with joint sparsity efficient. The $l_{2,1}$ -norms used is robust in outlier points. We make Transform FSNM into a hierarchical feature selection algorithm named HFSNM that is applied to hierarchical datasets. We use support vector machine to classify the classes after hierarchical feature selection.
- (2) HmRMR: mRMR [28] algorithm is an approximation of the best maximum dependency feature selection algorithm in theory, which maximizes mutual information between the selected features and the joint distribution of classification variables. HmRMR [15] has feature selection ability for hierarchical datasets. After finishing hierarchical feature selection, we need support vector machine to classify the classes.
- (3) HRelief: Relief [18] algorithm is a feature selection approach. Relief does subset search work instead of global search and gives different weights to different attributes. We modify Relief to HRelief for suiting hierarchical datasets. After feature selection process, support vector machine identifies the classes.
- (4) HFISHER: Fisher Score [9] has good discriminating capability for feature selection. Fisher Score finishes the feature selection process using full data, which is totally labeled. We modify FISHER Score feature selection to HFISHER for hierarchical datasets and use support vector machine to identify the labels.
- (5) LBRM [45]: LBRM algorithm is a hierarchical classification method based on a local Bayes risk minimization approach. The LBRM algorithm balances predicting risks for choose to go to a lower level or to finish the prediction process at nodes on the tree structure. The LBRM algorithm can decide not to go down to the lower level. This step can greatly avoid the inter-level error propagation in the hierarchical structure.
- (6) HCMP [16]: HCMP is a hierarchical classification algorithm method using the best k -th paths for another round classification at non-leaf nodes in a hierarchical classification. The HCMP algorithm first classifies the best several selection on each tree level with logistic regression method. The HCMP algorithm then uses random forest method to select the best node from these several nodes.
- (7) FSFHC [49]: FSFHC is a hierarchical classification algorithm based on a recursive regularization feature selection. FSFHC method uses the structural relationship of hierarchical parent-child as its hierarchical regularization strategy. Hilbert-Schmidt Independence

Criterion used in this method for measuring the distance of sibling nodes. Support vector machine algorithm is used for identifying classes after hierarchical feature selection.

4.3 Evaluation metrics

Classification accuracy (accuracy) Accuracy is an important evaluation metric to measure the classification performance of classifiers. We compare the predicted labels of the classifier with the correct labels, and the ratio of the correct number of predicted labels to the total number is accuracy.

Hierarchical measure (F_H) Hierarchical F_1 measure (F_H) is an important criterion for hierarchical classification. F_H measure considers the relationship between all ancestors and descendants of each class. F_H includes hierarchical precision (P_H), hierarchical recall (R_H). P_H , R_H and F_H are defined as follows.

$$P_H = \frac{|\hat{D}_{aug} \cap D_{aug}|}{|\hat{D}_{aug}|}, \quad (6)$$

$$R_H = \frac{|\hat{D}_{aug} \cap D_{aug}|}{|D_{aug}|}, \quad (7)$$

$$F_H = \frac{2 \cdot P_H \cdot R_H}{P_H + R_H}, \quad (8)$$

where $D_{aug} = D \cup anc(D)$, $\hat{D}_{aug} = \hat{D} \cup anc(\hat{D})$, $|\cdot|$ is the number of elements, D is the right label, \hat{D} is the predicted label, and $anc(D)$ is the parent label set of the right label to which the sample belongs.

5 Experimental results and discussion

In this section, we present the experimental results and discussion from three perspectives. In Section 5.1, we compare the running times of eight hierarchical classification methods to verify the efficiency of the CSHC algorithm. In Section 5.2, we compare the local classification accuracy at non-leaf nodes to discuss the impacts of cost sensitivity weights and the threshold strategy on hierarchical classification. In Section 5.3, we report the global hierarchical experimental results of the eight algorithms in terms of the two evaluation metrics to demonstrate the effectiveness of the proposed CSHC algorithm.

5.1 Running time comparison

Table 2 reports the performance of the eight hierarchical algorithms on the five datasets. Bold texts in Table 2 are used to emphasize the optimal calculation time of each

Table 2 Running time comparison of six algorithms on five different datasets (s)

Dataset (Avg.Rank)	ILSVRC57	CAR196	SUN	DD	F194
HFSNM (5.6)	2,088.4 (5)	1,385.3 (3)	11,788 (6)	132 (7)	804.7 (7)
HmRMR (5.8)	13,684.4 (6)	6,623.1 (8)	19,822 (7)	32.7 (5)	55.7 (3)
HRelief (6.8)	25,601 (7)	3,038.4 (7)	27,574 (8)	92.1 (6)	293.2 (6)
HFISHER (3)	1,260 (4)	2,738.4 (5)	5,439 (4)	0.39 (1)	2.24 (1)
LBRM (2.6)	447.4 (1)	477.6 (1)	1,322 (2)	16.5 (4)	164.5 (5)
HCMP (6.4)	5639 (5)	2839.8 (6)	9059.3 (5)	429.8 (8)	1018.9 (8)
FSFHC (2.6)	448.9 (2)	1938.7 (4)	2170 (3)	2.5 (2)	7.8 (2)
CSHC (2.6)	503 (3)	503.4 (2)	1,056.9 (1)	11.1 (3)	77.7 (4)

dataset. LBRM, HCMP, and CSHC are all capable of hierarchical classification. In processing large numbers of samples and features, the hierarchical classification system learned from hierarchical data and quickly completed classification of the ILSVRC57, CAR196, and SUN datasets. The four hierarchical algorithms with feature selection needed a long feature-selection process and took more than 1000 s to classify these three datasets. CSHC achieved the best result on the SUN dataset. HFISHER had the perfect performance on the DD and F194 datasets. The hierarchical classification with feature selection algorithms showed advantages on the two smaller datasets. These two datasets were much smaller than the others in terms of both samples and features. CSHC was a close second place and achieved classification in 11.1 and 77.7 s on these two datasets, respectively. The running time rank of the FSFHC algorithm on the five datasets was stable. HCMP needed two classification methods for data processing. It spent much more than 100 s on each of the five datasets. In terms of average rank, CSHC demonstrated good efficiency in handling large and small datasets.

5.2 Compared with traditional hierarchical classification at non-leaf nodes

In this section, we compare the CSHC algorithm with a traditional hierarchical classifier (THC) on non-leaf nodes of the DD and F194 datasets. Tables 3 and 4 introduce the non-leaf nodes, numbers of child nodes, proportions of each non-leaf node sample in the total samples, and classification accuracy at non-leaf nodes on the DD and F194 datasets.

Table 3 Information of non-leaf nodes and accuracy of two algorithms on DD dataset (%)

Node	α	β	α/β	$\alpha + \beta$
Child node	6	9	9	3
Proportion	16.38	27.87	39.8	15.94
THC	94.8	95.14	95.01	72.17
CSHC	95.3	94.35	93.01	78.03

The DD dataset is built as a three-level tree. There are four child nodes of the root node. Nodes α , β , α/β , and $\alpha+\beta$ are non-leaf nodes, called Nodes 1 to 4 for short. Nodes 1 and 4 are minority classes that have six and three child nodes, respectively. The proportions of their samples are significantly less than those of the other two nodes. The DD dataset includes 3,625 samples in total, but four of the classes have >300 samples. Most classes have about 100 samples. There are <50 samples in three of the classes. We know that CSHC’s classification accuracy of Nodes 1 and 4 is higher than that of THC. The classification accuracy of Node 4 by THC is 72%, while CSHC increases the accuracy to 78%. The accuracy of the other two non-leaf nodes is slightly reduced. In exchange, the accuracy of a minority class and the overall accuracy of this dataset are improved.

The F194 dataset is built as a three-level tree. There are seven non-leaf nodes on the second level. Nodes α , β , α/β , $\alpha+\beta$, Multi-domain, Membrane and cell surface (MCS) and Small-proteins are non-leaf nodes. We call them Nodes 1 to 7 for convenience. We know there is a big difference between the majority and minority classes from Table 4. Nodes 5 to 7 are the minority classes, which occupy a small number of samples and <10% of the total child nodes. The accuracy of the minority classes is much less than that of the majority. Node 5 has only 0.76% samples in this dataset. We can see more intuitively that the accuracy of the minority classes is significantly higher than that of THC after the cost-sensitive method and threshold strategy are applied. THC works well on several nodes, except for Nodes 5 and 6. The accuracies of Nodes 5 and 6 classified by THC are 1.49% and 26.79%, respectively. CSHC greatly improves the accuracies of Nodes 5 to 7 by 32% and 25% more than THC, respectively.

5.3 Experimental results and discussion on the whole hierarchical tree

In this section, we discuss the experimental results for the whole hierarchical tree. We first discuss the experimental results of two smaller datasets. Figures 5 and 6 show the experimental results in terms of two evaluation metrics

Table 4 Information of non-leaf nodes and accuracy of two algorithms on F194 dataset (%)

Node	α	β	α/β	$\alpha + \beta$	Multi-domain	MCS	Small-proteins
Child node	36	40	43	57	3	3	12
Proportion	17.28	23.37	29.74	23.24	0.76	0.95	4.66
THC	98.17	90.11	91.06	61.7	1.43	26.79	75.29
CSHC	98.1	89.9	89.49	61.75	33.51	51.98	88.03

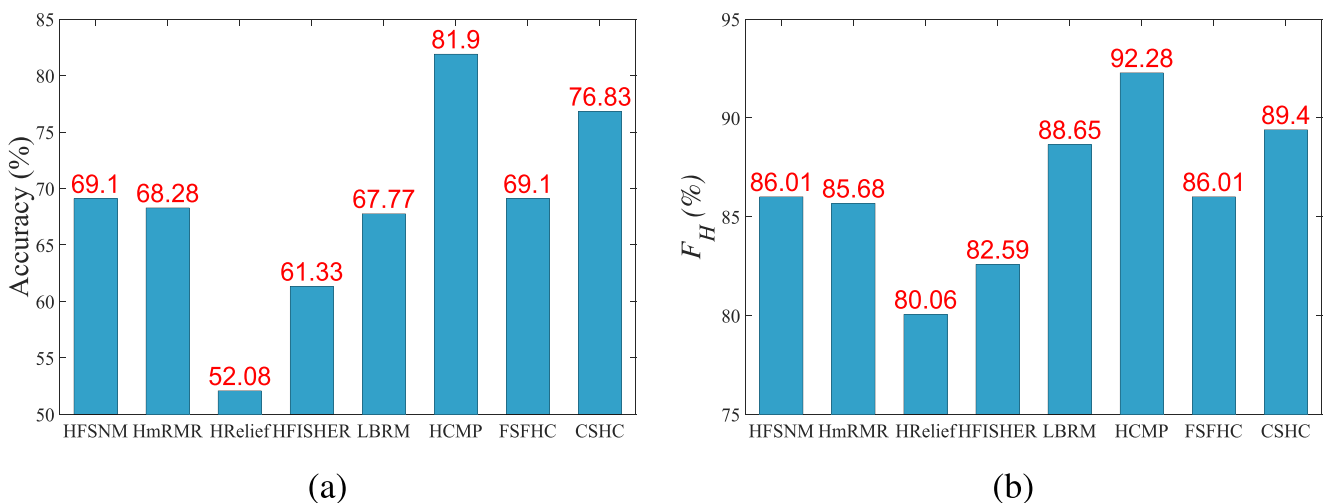
on the DD and F194 datasets. HCMP obtained the best results, with classification accuracies of 81.9% and 55.24%. CSHC showed a great performance advantage on the two datasets and is obviously more accurate than the other hierarchical classification with feature selection methods, achieving accuracies of 76.83% and 52.45% on the DD and F194 datasets, respectively. HCMP selects several better nodes on the basis of hierarchical classification and uses the random forest method to perform another classification. However, this causes HCMP to use much more running time than CSHC. The other methods' accuracies on the two datasets are less than 70% and 50%, respectively. LBRM behaves differently with the two datasets, producing an accuracy of 67.77% on the DD dataset but only 17.92% on the F194 dataset. HFSNM is second only to CSHC on the two datasets, with accuracies of 69.1% and 34.08%, respectively. The accuracy and F_H of HFSNM and FSFHC are exactly the same.

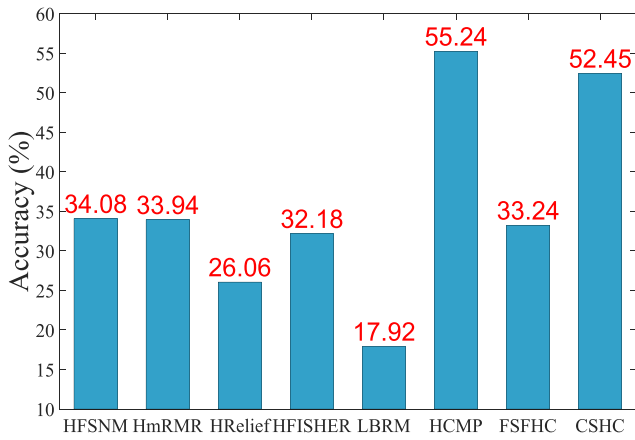
The CSHC, HCMP, and LBRM hierarchical classification methods are ahead of the others in terms of the F_H metric. CSHC obtained F_H of 89.4% and 78.74%, while LBRM obtained F_H of 88.65% and 75.52% on the DD and F194 datasets, respectively. CSHC uses the threshold strategy at non-leaf nodes and improves the accuracies of minority classes on the second level. The advantages of this strategy lie in its overall accuracy and hierarchical precision. Improvement in hierarchical precision makes CSHC

obtain a better value of the F_H metric. LBRM balances risks by considering going down or finishing the prediction process at each node. It is like making decisions at every non-leaf node. This process increases the hierarchical precision. HCMP takes advantage of two classification processes at non-leaf nodes, which reduce errors at non-leaf nodes and improve the F_H . LBRM does not have good accuracy on dataset F194, but the F_H of LBRM means that it classifies well at non-leaf nodes. HRelief is the worst hierarchical classification algorithm and only obtained F_H values of 80.06% and 68.85% on the DD and F194 datasets, respectively.

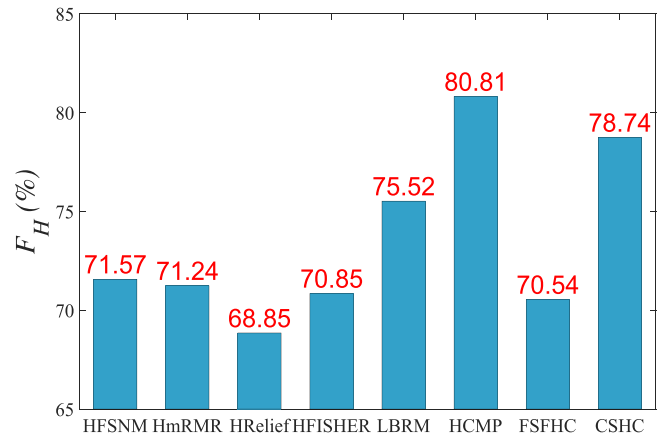
Figures 7, 8 and 9 show the experimental results of eight hierarchical classification algorithms on three large datasets. We observe the following:

The results of the four hierarchical classifiers that use feature selection algorithms, in terms of the two metrics, were averaged and are shown in Fig. 7a and b. The accuracies of HFISHER, FSFHC, and HmRMR were relatively close, at 85.01%, 85.53%, and 84.97%, respectively. The accuracy of CSHC reached 86.61%, which is 1% higher than the second-best classifier, FSFHC. The F_H of CSHC was 96.34%. The F_H of HFISHER, HFSNM, HmRMR, and HRelief were 95.8%, 95.77%, 95.81%, and 95.79%, respectively. LBRM reduced the gap with the others, with an F_H of 95.58%. The F_H of HCMP and HFSNM were the same, but the classification accuracy of

**Fig. 5** Accuracy and F_H on the DD dataset

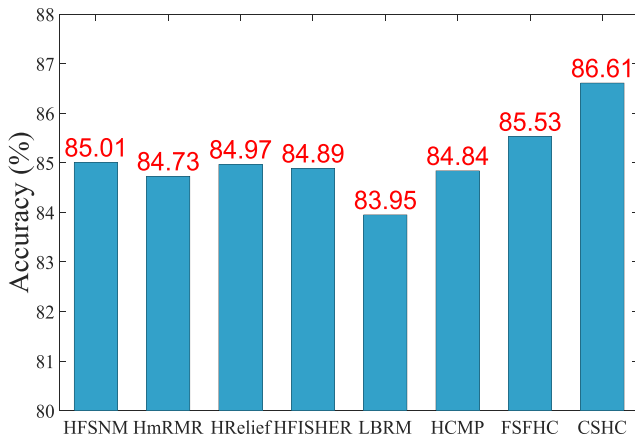


(a)

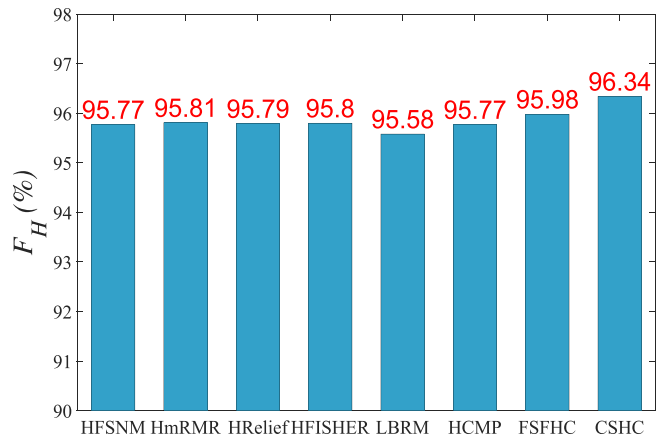


(b)

Fig. 6 Accuracy and F_H on the F194 dataset

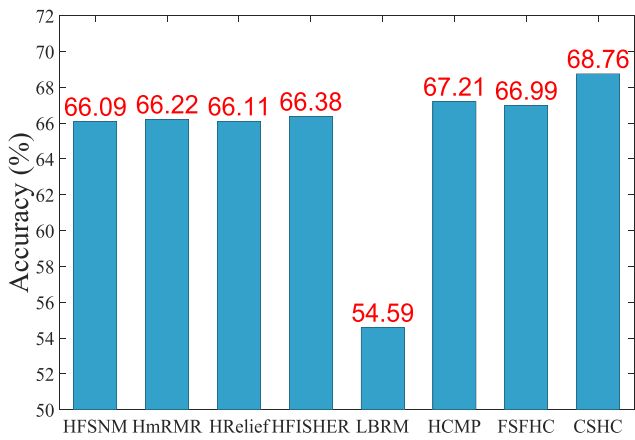


(a)

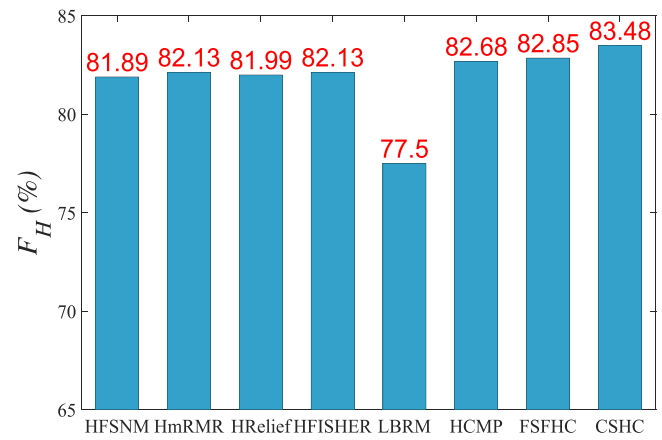


(b)

Fig. 7 Accuracy and F_H on the ILSVRC57 dataset



(a)



(b)

Fig. 8 Accuracy and F_H on the CAR196 dataset

HFSNM was a little higher. The HCMP result is in sixth place, as its two classification processes do not perform well on this dataset.

The features in the CAR196 dataset were extracted after deep learning. Figure 8 shows that the experimental results of the four hierarchical classifiers with feature selection in terms of the two evaluation metrics were similar, and only a little less than that of CSHC. HCMP obtained a 67.21% classification accuracy, which is at least 1.08% higher than that of the others. The accuracy of LBRM was nearly 10% lower than that of the others. In the same way, the performance of LBRM in terms of the F_H evaluation metric was poor. The four hierarchical classifiers with feature selection algorithms use good-quality features to achieve high accuracy and F_H values. The accuracy of FSFHC was lower than that of HCMP, but the F_H value of FSFHC was better. That shows that FSFHC performs better on the inter-levels. Its accuracy and F_H were 68.74% and 83.48%, respectively. Therefore, the experimental results of CSHC are very competitive.

Figure 9a and b reveal very different experimental results. We can see that the accuracy of LBRM was the lowest, but it was in third place in terms of the F_H metric. This indicates that LBRM has good performance in improving the precision of non-leaf nodes but not the accuracy of final leaf nodes. The result on the F194 and SUN datasets show that LBRM is good in terms of the F_H metric because of its balanced risk strategy at non-leaf nodes. The two evaluation metrics reflect the instability of LBRM and the advantage of CSHC. The experimental results for HCMP and CSHC are ahead of those of the other algorithms on the SUN dataset. The HCMP algorithm had good results because of its two classification processes. The results of FSFHC on this dataset are normal. CSHC identified this image classification dataset best, with accuracy and F_H values of

67.01% and 86.06%, respectively. The results of the four other hierarchical classification algorithms are consistent in terms of the two evaluation metrics.

The experimental results for the five imbalanced distribution datasets indicate that CSHC is more competitive than the other algorithms. CSHC can complete classification tasks quickly and maintain high accuracy. CSHC obtained the best accuracy on three large datasets and the second-best on two small datasets.

6 Conclusions and future work

We proposed a cost-sensitive hierarchical classification method based on multi-scale information entropy for addressing imbalanced distribution data. Unlike existing cost-sensitive algorithms, our algorithm uses the hierarchical structure as additional and supplemental information. In addition, we proposed an adaptive threshold method based on hierarchical data that does not require users to specify any parameters. Multi-scale information entropy is used in our hierarchical cost-sensitive learning method to address imbalanced distribution data. We compared our proposed algorithm with seven powerful hierarchical algorithms on five imbalanced distribution datasets. Our algorithm not only improves the accuracy of classification with few samples, but also ensures the overall accuracy of classification. For data with an imbalanced distribution, long-tail distributions create a new challenge because 80% of the samples are distributed in 20% of the majority classes and the minority only contain 20% of the samples. A good point to be considered is that we can get ideas from [1, 31] to build neural network algorithms for big data issues. In future, we will construct a neural network algorithm to address the long-tail distribution problems.

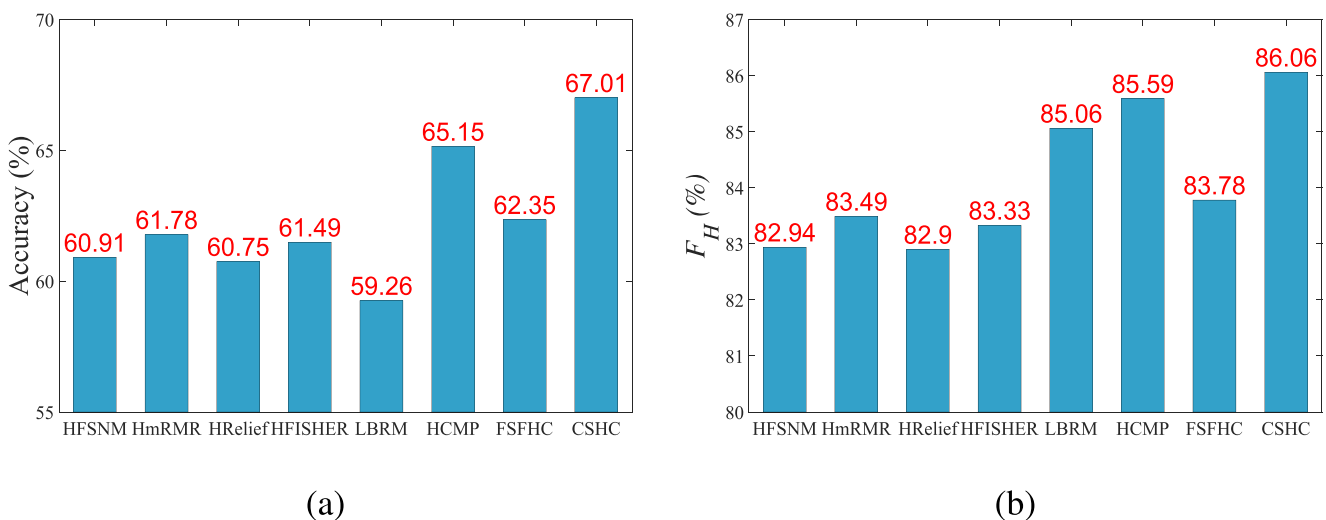


Fig. 9 Accuracy and F_H on the SUN dataset

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant No. 61703196, the Natural Science Foundation of Fujian Province under Grant No. 2018J01549, and the President's Fund of Minnan Normal University under Grant No. KJ19021.

References

- Ahmadian S, Khanteymooori A (2015) Training back propagation neural networks using asexual reproduction optimization. In: The 7th conference on information and knowledge technology, pp 1–6
- Braytee A, Wei L, Kennedy P (2016) A cost-sensitive learning strategy for feature extraction from imbalanced data. In: International conference on neural information processing
- Cai Z, Zhu W (2018) Multi-label feature selection via feature manifold learning and sparsity regularization. *Int J Mach Learn Cybern* 9(8):1321–1334
- Cao P, Zhao D, Zaiane O (2013) An optimized cost-sensitive SVM for imbalanced data learning. In: Pacific-Asia conference on knowledge discovery and data mining
- Castellanos F, Valero-Mas J, Calvo-Zaragoza J (2018) Oversampling imbalanced data in the string space. *Pattern Recognit Lett* 103:32–38
- Chen Y, Hu H, Tang K (2009) Constructing a decision tree from data with hierarchical class labels. *Exp Syst Appl* 36:4838–4847
- Dekel O, Keshet J, Singer Y (2004) Large margin hierarchical classification. In: International conference on machine learning
- Ding C, Dubchak I (2001) Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics* 17(4):349–358
- Duda R, Hart P, Stork D (2001) *Pattern classification*. Wiley
- Fan J, Gao Y, Luo H, Jain R (2008) Mining multilevel image semantics via hierarchical classification. *IEEE Trans Multimed* 10(2):167–187
- Fan J, Zhang J, Mei K, Peng J, Gao L (2015) Cost-sensitive learning of hierarchical tree classifiers for large-scale image classification and novel category detection. *Pattern Recognit* 48(5):1673–1687
- Fawcett T, Provost F (1997) Adaptive fraud detection. *Data Min Knowl Discov* 1(3):291–316
- Feng F, Li K, Shen J (2020) Using cost-sensitive learning and feature selection algorithms to improve the performance of imbalanced classification. *IEEE Access* 10(99):1–12
- Ghatasheh N, Faris H, Altaharwa I (2020) Business analytics in telemarketing: cost-sensitive analysis of bank campaigns using artificial neural networks. *Appl Ences* 10(7):2581–2592
- Grimaudo L, Mellia M, Baralis E (2012) Hierarchical learning for fine grained internet traffic classification. In: International wireless communications and mobile computing conference
- Guo S, Zhao H (2020) Hierarchical classification with multi-path selection based on granular computing. *Artif Intell Rev* (1)1–23
- Khan S, Hayat M, Bennamoun M, Sohel F, Togneri R (2018) Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Trans Neural Netw Learn Syst* 29(8):3573–3587
- Kira K, Rendell L (1992) A practical approach to feature selection. In: International workshop on machine learning
- Krause J, Stark M, Deng J (2013) Li, F: 3D object representations for fine-grained categorization. In: International IEEE workshop on 3D representation and recognition
- Lin W, Tsai C, Hu Y et al (2017) Clustering-based undersampling in class-imbalanced data. *Inf Sci* 17(26):409–419
- Ling C, Sheng S, Qiang Y (2006) Simple test strategies for cost-sensitive decision trees. *IEEE Trans Knowl Data Eng* 8(18):1055–1067
- Liu J, Hu Q, Yu D (2008) A weighted rough set based method developed for class imbalance learning. *Inf Sci* 178(4):1235–1256
- Liu X, Wu J, Zhou Z (2009) Exploratory undersampling for class-imbalance learning. *IEEE Trans Syst Man Cybern Part B* 39(2):539–550
- Lu J, Tan Y (2010) Cost-sensitive subspace learning for human age estimation. In: Proceedings of the international conference on image processing
- Min F, He H, Qian Y et al (2011) Test-cost-sensitive attribute reduction. *Information Sciences An International Journal* 181(22):4928–4942
- Nakano F, Pinto W, Pappa G, Cerri R (2017) Top-down strategies for hierarchical classification of transposable elements with neural networks. In: International joint conference on neural networks
- Nie F, Huang H, Xiao C, Ding C (2010) Efficient and robust feature selection via joint $l_{2,1}$ -norms minimization. In: International conference on neural information processing systems
- Peng H, Long F, Ding C (2005) Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Mach Intell* 27(8):1226–1238
- Qing T, Wu G, Wang F (2005) Posterior probability support vector machines for unbalanced data. *IEEE Trans Neural Netw* 16(6):1561–1573
- Sahin Y, Bulkan S, Duman E (2013) A cost-sensitive decision tree approach for fraud detection. *Exp Syst Appl* 40(15):5916–5923
- Sajad A, Ali K (2019) Evolving artificial neural networks using butterfly optimization algorithm for data classification. In: International conference on neural information processing, pp 596–609
- Sandrine D, Jane F (2002) A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biol* 3(7):1–21
- Sayed J, Sajad A, Abbas K et al (2020) Neuroevolution-based autonomous robot navigation: a comparative study. *Cogn Syst Res* 62:35–43
- Sheng S, Ling C, Ni A, Zhang S (2006) Cost-sensitive test strategies. In: Conference on AAAI Press
- Sun A, Lim E (2001) Hierarchical text classification and evaluation. In: IEEE international conference on data mining
- Sun Y, Kamel M, Wong A, Wang Y (2007) Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognit* 40(12):3358–3378
- Thai-Nghe N, Gantner Z, Schmidt L (2010) Cost-sensitive learning methods for imbalanced data. In: International joint conference on neural networks
- Tuo Q, Zhao H, Hu Q (2019) Hierarchical feature selection with subtree based graph regularization. *Knowl-Based Syst* 163:996–1008
- Wang C, Wang Y, Shao M, Qian Y, Chen D (2009) Fuzzy rough attribute reduction for categorical data. *IEEE Trans Fuzzy Syst* pp(99):1–12
- Wang S, Zhu W (2018) Sparse graph embedding unsupervised feature selection. *IEEE Trans Syst Man Cybern Syst* 48(3):329–341

41. Wang C, Huang Y, Shao M, Hu Q, Chen D (2019) Feature selection based on neighborhood self-information. *IEEE Trans Cybern pp(99)*:1–12
42. Wei L, Liao M, Gao X, Zou Q (2015) An improved protein structural prediction method by incorporating both sequence and structure information. *IEEE Trans Nanobiosci* 14(4):339–349
43. Xiao J, Hays J, Ehinger K, Oliva A, Torralba A (2010) Sun database: large-scale scene recognition from abbey to zoo. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, vol 23, pp 3485–3492
44. Yu X, Liu J, Keung J (2020) Improving ranking-oriented defect prediction using a cost-sensitive ranking SVM. *IEEE Trans Reliab* 69(1):139–153
45. Yu W, Hu Q, Zhou Y, Hong Z, Qian Y, Liang J (2017) Local bayes risk minimization based stopping strategy for hierarchical classification. In: *IEEE international conference on data mining*
46. Zadrozny B, Langford J, Abe N (2003) Cost-sensitive learning by cost-proportionate example weighting. In: *IEEE international conference on data mining*
47. Zhang Y, Zhou Z (2010) Cost-sensitive face recognition. *IEEE Trans Pattern Anal Mach Intell* 10(32):1758–1769
48. Zhao H, Hu Q, Wang P (2017) Hierarchical feature selection with recursive regularization. In: *International joint conference on artificial intelligence*, pp 3483–3489
49. Zhao H, Hu Q, Zhu P et al (2019) A recursive regularization based feature selection framework for hierarchical classification. *IEEE Trans Knowl Data Eng PP(99)*:10–23
50. Zhou Y, Hu Q, Yu W (2018) Deep super-class learning for long-tail distributed image classification. *Pattern Recognit* 80:118–128

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Weijie Zheng is currently a M.S. student with the Fujian Key Laboratory of Granular Computing and Application, Minnan Normal University, Zhangzhou, China. His current research interests on data mining and machine learning for hierarchical classification.



Hong Zhao received the Ph.D degree from Tianjin University, Tianjin, China, in 2019. She received her M.S. degree from Liaoning Normal University, Dalian, China, in 2006. 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. She has authored over 40 journal and conference papers in the areas of granular computing based machine learning and cost-sensitive learning. Her current research interests include rough sets, granular computing, and data mining for hierarchical classification.