Expert Systems With Applications 100 (2018) 157–164

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Hierarchy construction and text classification based on the relaxation strategy and least information model

Yongping Du[a,*], Jingxuan Liu[a], Weimao Ke[b], Xuemei Gong[b]

[a] *Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[b] *College of Computing and Informatics, Drexel University, Philadelphia 19104, USA*

ARTICLE INFO

Article history:
Received 28 November 2017
Revised 20 January 2018
Accepted 1 February 2018

Keywords:
Hierarchy classification
Relaxation strategy
Least Information Theory
Term weighting

ABSTRACT

Hierarchical classification is an effective approach to categorization of large-scale text data. We introduce a relaxed strategy into the traditional hierarchical classification method to improve the system performance. During the process of hierarchy structure construction, our method delays node judgment of the uncertain category until it can be classified clearly. This approach effectively alleviates the 'block' problem which transfers the classification error from the higher level to the lower level in the hierarchy structure. A new term weighting approach based on the Least Information Theory (LIT) is adopted for the hierarchy classification. It quantifies information in probability distribution changes and offers a new document representation model where the contribution of each term can be properly weighted. The experimental results show that the relaxation approach builds a more reasonable hierarchy and further improves classification performance. It also outperforms other classification methods such as SVM (Support Vector Machine) in terms of efficiency and the approach is more efficient for large-scale text classification tasks. Compared to the classic term weighting method TF*IDF, LIT-based methods achieves significant improvement on the classification performance.

1. Introduction

The task of text classification is to assign a predefined category to a free text document. With more and more textual information available online, hierarchical organization of text documents is becoming increasingly important to manage the data. The research on automatic classification of documents to the categories in the hierarchy is needed.

Most of the classifiers make the decision in the same flat space. Classification performance degrades quickly with larger scale data sets and more categories, especially in terms of the classification time. On the other hand, a hierarchical classification method organizes all of the categories into a tree like structure and trains a classifier on each node in the hierarchy. The classification process begins from the root of the tree until it reaches the leaf node which denotes the final category for the document.

The hierarchies are represented as binary trees mostly. During the hierarchical classification process, the document to be classified starts from the root and the next direction is determined by each node classifier. Finally, the leaf being reached will give the decision to its category label. However, there exists a 'blocking' problem during the process. The error that has occurred in the upper node classifier cannot be corrected by the lower node classifier. The 'blocking' problem may result in weaker performance compared to the non-hierarchical classification method. The advantage of hierarchy classification method is higher efficiency which is significant in large scale data set.

In order to improve the hierarchical classification performance, we introduce the relaxation strategy idea during the process of hierarchy construction and further propose the hierarchical classification approach based on it. The method delays the uncertain category decision until it can be classified definitely, thereby alleviating the impact of the 'blocking' problem. We give the experiment on the Reuters Corpus Volume 1(RCV1). The result denotes that our method can build a more rational category hierarchy and improve the performance of traditional hierarchy classification. Especially, the approach has higher time efficiency than other classifiers such as Support Vector Machine.

Another contribution of this work is in term weighting and documentation representation. The classic TF*IDF has been widely used for term weighting and document representation in text clustering and classification tasks (Liu, Liu, Chen, & Ma, 2003; Yang & Pedersen, 1997). Least Information Theory (LIT) extends Shannon's information theory to accommodate a non-linear relation between

[*] Corresponding author.
E-mail addresses: ypdu@bjut.edu.cn (Y. Du), Liujx99@emails.bjut.edu.cn (J. Liu), wk@drexel.edu (W. Ke), xuemeigong@drexel.edu (X. Gong).

https://doi.org/10.1016/j.eswa.2018.02.003
0957-4174/© 2018 Elsevier Ltd. All rights reserved.

information and uncertainty and offers a new way of modeling for term weighting and document representation (Ke, 2015). It establishes a new basic information quantity and provides insight into how terms can be weighted based on their probability distributions in documents vs. in the collection. We adopt the LIT for term weighting during hierarchical classification and it achieves significant performance improvement over classic TF*IDF.

2. Related work

It is important to build the rational category hierarchy and there are two common ways to implement this, including the Top-Down and Bottom-Up approaches. Liu, Yi, and Chia (2005) present a method to build up a hierarchical structure from the training dataset and uses the K-Means clustering algorithm to divide the category set. The hierarchical structure of the SVM classification tree manifests the interclass relationships among different classes. Chen, Crawford, and Ghosh (2004) propose the segmentation approach using the maximum division strategy. It presents a new approach called HSVM (Hierarchical Support Vector Machines) to address multiclass problems. The method solves a series of max-cut problems to hierarchically and recursively partition the set of classes into two-subsets. The way of Bottom-Up cannot guarantee the separability of the category node set and the Top-Down approach is more commonly used.

Most hierarchy building methods organize the categories into a tree structure and usually the hierarchies are represented as binary trees which means that at each node a binary decision is made on which of the two subtrees to choose (Griffin & Perona, 2008). Marcin (2008) proposes a new idea which allows the child node has more than one parent node, and all the categories are organized as the DAG (Directed Acyclic Graph) structure. This idea has been used in the field of image classification and shows strong performance.

There are also some works that are focused on the Hierarchical Multi-label Classification. Each parent node is divided into multiple child nodes and the process is continued until each child node represents only one class. Zhang, Shah, and Kakadiaris (2017) consider the structural information embedded in the class hierarchy and uses it to improve the hierarchical classification performance. Bengio, Weston, and Grangier (2010) introduces an approach for fast multi-class classification by learning label embedding trees and it outperforms other tree-based or embedding approaches.

The traditional text classification approaches often require labeled data for learning classifiers, which is extremely expensive when applied to large-scale data involving thousands of categories. Viet (2011) takes advantage of the ontological knowledge for large-scale hierarchical text classification which does not require any labeled data. The classifier gets a reasonable performance. Pavlinek and Podgorelec (2017) presents the Self-Training LDA method for text classification in a semi-supervised manner with representations based on topic models.

The hierarchical classification approach decomposes the multi-class classification problem into different sub-task, and every node classifier solves the sub-task separately. The linear classifier (Deng, Satheesh, Berg, & Li, 2011), Bayesian Network (Wang, Wang, & Xie, 2011) and Support Vector Machine (Gao & Koller, 2011; Griffin & Perona, 2008) are used as the node classifier.

Another important aspect of this research is on feature selection and weighting for classification. In text clustering and classification research, TF*IDF has been extensively used for term weighting and document representation (Liu et al., 2003; Yang & Pedersen, 1997; Zhang, Wang, & Si, 2011). While term frequency (TF) indicates the degree of a document's association with a term, inverse document frequency (IDF) is the manifestation of a term's specificity, key to determine the term's value toward weighting and relevance ranking (Jones, 2004). Chen, Zhang, Long, and Zhang (2016) propose a new term weighting scheme TF-IGM (term frequency & inverse gravity moment) which incorporates a new statistical model to precisely measure the class distinguishing power of a term. Deepak, Kesari, and Priyanka (2017) propose a novel Variable Global Feature Selection Scheme (VGFSS) to select a variable number of features from each class based on the distribution of terms in the classes. While many classification algorithms have been developed, TF*IDF and its variations remain the de facto standard for term weighting in classification.

In IR (Information Retrieval), information and probability theories have provided important guidance to the development of classic techniques such as probabilistic retrieval and language modeling (Robertson & Zaragoza, 2009). The probabilistic retrieval framework provides an important theoretical ground to IDF weights (Robertson, 2004). IDF resembles the entropy formula in Shannon's information theory and several works have attempted to justify IDF from an information-theoretic view. IDF can be interpreted as Kullback–Leibler (KL) information (relative entropy) between term probability distributions in a document and in the collection (Aizawa, 2000). KL divergence measures information for discrimination between two probability distributions by quantifying the entropy change in a non-symmetric manner (Kullback & Leibler, 1951).

In the KL information view of IDF, the asymmetry of KL and infinite information it quantifies in special cases have undesirable consequences in the text classification context. From an information-centric view, Ke (2015) developed a new model for term weighting and document representation. By quantifying the amount of semantic information required to explain probability distribution changes, the proposed Least Information Theory (LIT) offers a new measure through which terms can be weighted based on their probability distributions in documents vs. in the collection. Several term weighting schemes such as LI Binary (LIB) and LI Frequency (LIF) were derived and experimented for text clustering. In this research, based on the notion of mutual information and the new LIT theory, we propose Least Information Gain for feature selection and combinations of other LIT-based methods for hierarchy construction and classification. We are interested in understanding the effectiveness of LIT in hierarchical classification tasks.

3. Hierarchy construction with relaxation strategy

3.1. Relaxation method

The category set will be divided into two subsets recursively to build a hierarchical structure that contains *n* categories. K-Means clustering algorithm is adopted to get the two clusters on the text data set, and it will help to determine which node the category belongs to.

As shown in Fig. 1, the aim is to divide the root node $\boldsymbol{S}$, which contains category ***A, B*** and ***C***, into two subsets referred to $\boldsymbol{S_L}$ and $\boldsymbol{S_R}$ respectively.

As an example, there are 30 training documents *A01,…, A10, B01,…, B10, C01,…, C10* in the root Node $\boldsymbol{S}$ and they belong to category ***A, B,*** and ***C*** respectively. These documents are clustered into two sets by K-Means with label $+1$ and -1. We find that most of the documents in category ***A*** are labeled as $+1$ and so the category ***A*** is assigned to $\boldsymbol{S_L}$. Similarly, the category ***C*** is assigned to $\boldsymbol{S_R}$. For category ***B***, there are 6 documents labeled as $+1$ and 4 documents labeled as -1. It is uncertain to decide which node the category ***B*** belongs to. We delay the decision to the next lower level by assigning the category ***B*** to $\boldsymbol{S_L}$ and $\boldsymbol{S_R}$ simultaneously. This relaxation idea will be used during the process of hierarchy construction.

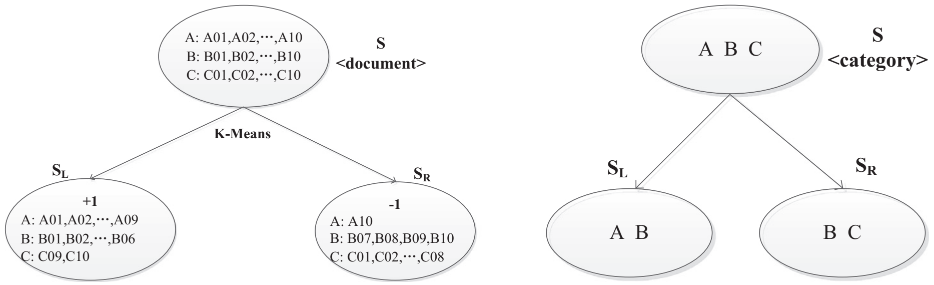


Fig. 1. Node division sample by K-means.

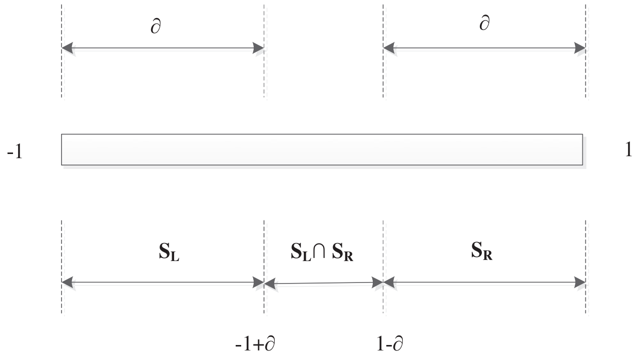


Fig. 2. Impact to the category division by parameter **∂**.

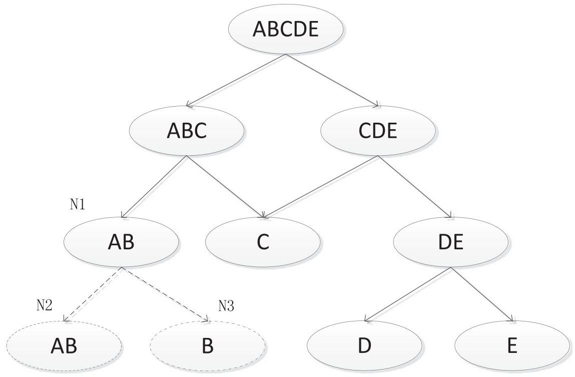


Fig. 3. Hierarchy structure sample.

Here, $\boldsymbol{s_i}$ denotes the category in the node set $\boldsymbol{S}$. $\boldsymbol{D}$ denotes document set of the node $\boldsymbol{S}$ and $\boldsymbol{d_i}$ represents the document set of category $\boldsymbol{s_i}$. Also $\boldsymbol{d_{ik}}$ denotes the $\boldsymbol{k}$**th** document in $\boldsymbol{d_i}$.

$$s_i \in S,\ \ i = 1, 2, \ldots, |S| \tag{1}$$

$$d_i \in D,\ \ i = 1, 2, \ldots, |S| \tag{2}$$

$$d_{ik} \in d_i,\ \ i = 1, 2, \ldots, |S|,\ \ k = 1, 2, \ldots, |d_i| \tag{3}$$

During the hierarchy construction process, each category $\boldsymbol{s_i}$ in $\boldsymbol{S}$ will be assigned to different child node $\boldsymbol{S_L}$ or $\boldsymbol{S_R}$. K-Means is implemented on the document set in $\boldsymbol{S}$ and k is set to 2. Each document $\boldsymbol{d_{ik}}$ in $\boldsymbol{d_i}$ will be assigned the value of $+1$ or -1 which is labeled as $\boldsymbol{p_{ik}}$. We compute the $\boldsymbol{q_i}$ value for each category $\boldsymbol{s_i}$ according to the $\boldsymbol{p_{ik}}$ value of each document $\boldsymbol{d_{ik}}$ by Eq.(4).

$$q_i = \frac{1}{|d_i|} \sum_{k=1,2,\ldots,|d_i|} p_{ik},\ \ p_{ik} = \{+1,\ -1\} \tag{4}$$

Finally, category $\boldsymbol{s_i}$ will be assigned to the child node $\boldsymbol{S_L}$ or $\boldsymbol{S_R}$ by the $\boldsymbol{q_i}$ value according to Eq.(5).

$$\begin{cases} s_i \in S_L,\ \ if\ q_i < -1 + \partial \\ s_i \in S_R,\ \ if\ q_i > 1 - \partial \\ s_i \in S_L\ and\ s_i \in S_R,\ \ if\ \ -1 + \partial \le q_i \le 1 - \partial \end{cases} \tag{5}$$

Here, parameter **∂** is the relaxation factor. The smaller value of **∂** will result in the larger size of $S_L \cap S_R$ and otherwise the size of $S_L \cap S_R$ is smaller with larger value of **∂**, which is shown in Fig. 2.

The category set $\boldsymbol{S}$ will be divided into two subsets from top to down recursively until there is only one category in the node or the node is inseparable. The hierarchy structure built by the relaxation strategy is no longer the tree like structure and it is organized as the DAG (Directed Acyclic Graph) structure, which allows the child node has more than one parent node. For example in Fig. 3, the node 'ABC' and 'CDE' contain the same category $\boldsymbol{C}$ which has two parent nodes. In addition, leaf node ***N1 'AB'*** cannot be divided by K-Means algorithm because its left child is the same with its parent.

3.2. Algorithm for hierarchy construction

The algorithm of text hierarchy structure construction based on the relaxation strategy is shown in Table 1. Relaxation factor **∂** is used to control the division of the category set and it provides a better relaxation by delaying the decisions for a subset of confusing classes.

4. Hierarchical classification based on the Least Information Theory for term weighting

4.1. Least information model for term weighting

The Least Information Theory (LIT) measures the distance between two probability distributions in a way different from Kullback–Leibler (KL) divergence (Lin, 1991). It establishes a new basic information quantity and provides insight into how terms can be weighted based on their probability distributions in documents vs. in the collection.

We apply the proposed Least Information Theory (Ke, 2015) to term weighting and document representation. A text document can be viewed as a set of terms with probabilities of occurrence. The larger amount least information is needed to explain a term's probability in a document (vs. in the collection), the more heavily the term should be weighted to represent the document (Amati and Rijsbergen, 2002).

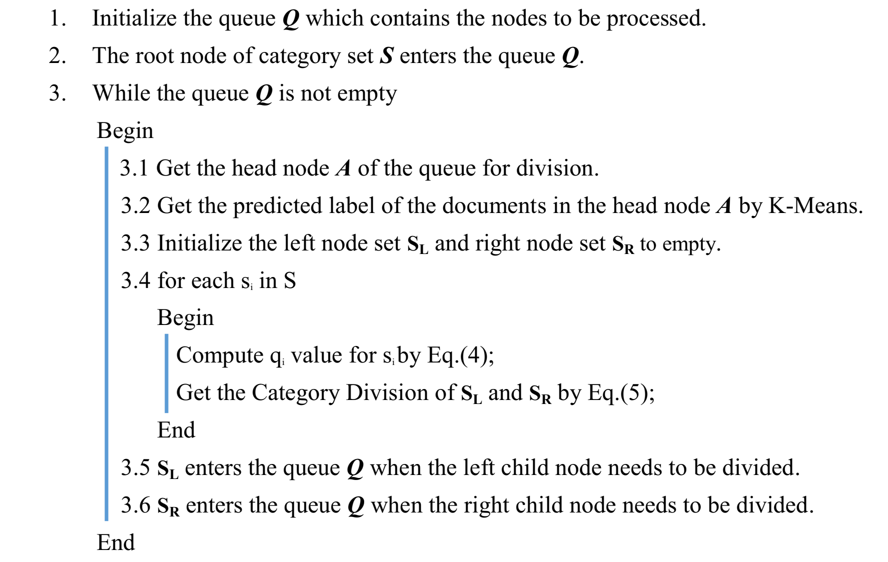
The information entropy $\boldsymbol{g_i}$ for variable $\boldsymbol{i}$ is defined as a function of its probability:

$$g_i = p_i(1 - lnp_i) \tag{6}$$

LIT has been applied in text clustering and information retrieval for term weighting and this method demonstrates the strong performance compared to classic TF*IDF (Ke, 2015).

Table 1
Hierarchy structure construction algorithm based on relaxation strategy.

Algorithm 1. Hierarchy Structure Construction based on Relaxation Strategy
Input: Category Set ***S***, Document set ***D*** with category labels, Relaxation factor ∂.
Output: Hierarchy Structure on set ***S***.



```
1.  Initialize the queue $Q$ which contains the nodes to be processed.
2.  The root node of category set $S$ enters the queue $Q$.
3.  While the queue $Q$ is not empty
    Begin
      3.1 Get the head node $A$ of the queue for division.
      3.2 Get the predicted label of the documents in the head node $A$ by K-Means.
      3.3 Initialize the left node set $\mathbf{S_L}$ and right node set $\mathbf{S_R}$ to empty.
      3.4 for each $s_i$ in S
          Begin
            Compute $q_i$ value for $s_i$ by Eq.(4);
            Get the Category Division of $\mathbf{S_L}$ and $\mathbf{S_R}$ by Eq.(5);
          End
      3.5 $\mathbf{S_L}$ enters the queue $Q$ when the left child node needs to be divided.
      3.6 $\mathbf{S_R}$ enters the queue $Q$ when the right child node needs to be divided.
    End
```

We adopt two basic quantities for document representation by the use of LIT. They are LI Binary (LIB) and LI Frequency (LIF) which are introduced in the following. And also the LI Gain (LIG) method is used for feature selection.

4.1.1. Least Information Binary Model (LIB)

LI Binary (LIB) quantifies information due to the observation of a term's binary occurrence in a document.

Given the definition of g_i in Eq. (6), the least amount of information in term t_i from observing document d can be computed by Eq. (7).

$$\mathrm{LIB}(t_i,\mathrm{d}) = \mathrm{g}(t_i|\mathrm{d}) - \mathrm{g}(t_i|\mathrm{C}) = \mathrm{g}(t_i|\mathrm{d}) - \frac{n_i}{N}\left(1 - ln\frac{n_i}{N}\right) \tag{7}$$

where n_i is the number of documents containing the term t_i and N is the total number of documents in the collection ***C***. $p(t_i\ |C) = n_i\ /N$ denote the probability of term t_i occurring in a randomly picked document in collection ***C***.

The larger of the LIB, the more information the term contributes to the document and it should be weighted more heavily in the document representation. The quantity depends on the observation of term t_i in the document: $g(t_i\ |d)$ is 1 when t_i appears in document d and 0 if otherwise. The LIB value is computed as Eq.(8).

$$\mathrm{LIB}(t_i,\mathrm{d}) = \begin{cases} 1 - \frac{n_i}{N}\left(1 - ln\frac{n_i}{N}\right) & t_i \in d \\ -\frac{n_i}{N}\left(1 - ln\frac{n_i}{N}\right) & t_i \ \notin d \end{cases} \tag{8}$$

4.1.2. Least Information Frequency Model (LIF)

LI Frequency (LIF) measures information for the observation of a randomly picked term from the document. The term frequency is used to model least information. It explains the change from the term's probability distribution in the collection to its distribution in the document.

For a document collection ***C***, the probability of a term t_i randomly selected from the collection ***C*** can be estimated by $p(t_i\ |C) = F_i\ /L$, where F_i is the total number of occurrences of term t_i in collection C and L is the overall length of C. According the definition of g_i in Eq.(6), the LIF value is measured by Eq.(9).

$$\mathrm{LIF}(t_i,\mathrm{d}) = \mathrm{g}(t_i|\mathrm{d}) - \mathrm{g}(t_i|\mathrm{C}) = \frac{tf_{i,d}}{L_d}\left(1 - ln\frac{tf_{i,d}}{L_d}\right) - \frac{F_i}{L}\left(1 - ln\frac{F_i}{L}\right) \tag{9}$$

When a specific document d is observed, the probability of picking term t_i from this document can be estimated by $p(t_i\ |d) = tf_{i,d}\ /\ L_d$, where $tf_{i,d}$ is the number of occurrences of term t_i in document d and L_d is the length of the document.

4.1.3. Least Information Gain Model (LIG)

Information gain is frequently used as a term weighting method and it measures the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document (Gong, 2015). It is to measure the difference between two probability distributions. We define the set S={s_i, 1 ≤ i ≤ n} which contains the categories in the target space. The information gain of term t is defined as Eq. (10).

$$\mathrm{IG}(\mathrm{t}) = \sum_{i=1}^{n} p(s_i \wedge t)\log\frac{p(s_i \wedge t)}{p(s_i)p(t)} + \sum_{i=1}^{n} p\left(s_i \wedge \bar{t}\right) log\frac{p\left(s_i \wedge \bar{t}\right)}{p(s_i)p\left(\bar{t}\right)} \tag{10}$$

Based on the ideas of Information Gain (IG) and the Least Information Theory (LIT), Least Information Gain (LIG) of term ***t*** is calculated by:

$$\begin{aligned}\mathrm{LIG}(\mathrm{t}) &= \sum_{i=1}^{n}\left|g(s_i \wedge t) - g(s_i t)\right| + \sum_{i=1}^{n}\left|g(s_i \wedge \bar{t}) - g(s_i, \bar{t})\right| \\ &= \sum_{i=1}^{n}\left|p(s_i \wedge t)(1 - lnp(s_i \wedge t)) - p(s_i)p(t)(1 - \ln(p(s_i)p(t)))\right| \\ &= \sum_{i=1}^{n}\left|p(s_i \wedge \bar{t})\left(1 - lnp(s_i \wedge \bar{t})\right) - p(s_i)p\left(\bar{t}\right)\left(1 - \ln\left(p(s_i)p\left(\bar{t}\right)\right)\right)\right|\end{aligned} \tag{11}$$

Here, $p(s_i)$ denotes the probability of a randomly picked document belonging to category $\boldsymbol{s_i}$. ***p(t)*** is the probability of a document

Table 2
Performance impact by the ∂ value.

Data	Evaluation	$\partial = 0.4$	$\partial = 0.6$	$\partial = 0.8$	$\partial = 1.0$
RCV1_10	Precision	0.7662	**0.7816**	0.7742	0.7712
	Recall	0.7612	**0.7633**	0.7577	0.7416
	F1	0.7639	0.7685	**0.7699**	0.7561
RCV1_15	Precision	0.7758	**0.7917**	0.7832	0.7866
	Recall	**0.7556**	0.7520	0.7467	0.7356
	F1	0.7621	**0.7645**	0.7564	0.7467
RCV1_20	Precision	0.7363	**0.7503**	0.7339	0.7288
	Recall	0.6860	**0.6930**	0.6841	0.6702
	F1	0.7100	**0.7210**	0.7091	0.6982
RCV1_25	Precision	0.6962	**0.7111**	0.6950	0.6878
	Recall	0.6520	**0.6555**	0.6510	0.6479
	F1	0.6725	**0.6736**	0.6720	0.6671
RCV1_30	Precision	0.6689	**0.6852**	0.6779	0.6630
	Recall	0.6290	**0.6442**	0.6348	0.6281
	F1	0.6480	0.6547	**0.6559**	0.6449

containing term $\boldsymbol{t}$. p($s_i \wedge$t) denotes the probability of a document which contains term $\boldsymbol{t}$ and also belongs to category $\boldsymbol{s_i}$. The greater of the LIG value for term $\boldsymbol{t}$, the more information it carries to reveal the content of the category. The LIG method has been used for cluster labeling, where it shows strong performance (Gong & Ke, 2015).

4.2. Classifier training on the node of the hierarchical structure

The hierarchical text classification process will start from the root of the hierarchy constructed in Section 3. The SVM classifier is trained for each node in the hierarchy. The documents in the set of $\boldsymbol{S_L}$ will be used as the positive samples and the documents in the set of $\boldsymbol{S_R}$ will be used as the negative samples. For each document in the test data set, it will be assigned to the left child node when the classifier predicts the category of $+1$, otherwise it will be assigned to the right child node.

The classification process in the hierarchy will continue until it reaches a leaf node. The category of the leaf node is assigned to the test document when the leaf node contains only one category. On the contrary, the classifier in the leaf node will be used to determine the final category label when the node contains more than one category.

5. Experimental evaluation

5.1. Data sets

We conducted the experiments on the Reuters Corpus Volume 1 (RCV1-v2) data set. The collection contains 804,414 newswire stories made available by Reuters. RCV1-v2 is a corrected version of the original collection, in which documents were manually assigned to a hierarchy of 103 categories. We select 10, 15, 20, 25 and 30 categories respectively to build the different size data sets, labeled as RCV1_10, RCV1_15, RCV1_20, RCV1_25 and RCV1_30. There are 500 documents selected randomly for each category and 350 documents are used as the training data. The system performance is evaluated by precision, recall and F1 score.

5.2. Evaluation result

5.2.1. Performance impact by the relaxation factor ∂

The hierarchy constructed is varied with different ∂ value described in Section 3 and the system performance for classification is also affected. The experimental results are shown in Table 2.

It can be concluded that the hierarchy built with the ∂ value of 0.6 gets the better system performance mostly on different data sets. The relaxation strategy does not work with the ∂ value of 1.0 which divides the parent node into two child nodes with the empty intersection set. The experimental result shows that this kind of hierarchy constructed without relaxation has poor classification performance. On the other hand, there are more overlapped categories between the child nodes with a smaller ∂ value, leading to the increased height of the hierarchy constructed. It results in the serious 'blocking' problem and poor performance.

5.2.2. Impact of the term weighting method by LIT

We use the Chi-square method for feature selection in combination with various term weighting methods such as TF*IDF, LIB, LIF and LIB*LIF. At the same time, different classifiers are also implemented to verify the effectiveness of the new term weighting methods. Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem. It requires only a small number of training data to estimate the parameters necessary for classification. Random Forests are an ensemble learning method, which operate by constructing a multitude of decision trees at training time and outputting the class. Bayes network is a probabilistic graphical model which has great advantages in solving the problems caused by the uncertainty.

The experimental results are shown in Table 3. Here, our relaxation hierarchy methods are denoted by RH_1, RH_2, RH_3 and RH_4 respectively with different ∂ values.

As shown in Table 3, we find that the system performance achieves the improvement mostly by the use of LIT method for term weighting, except for the Naive Bayes classifier on LIB. RH method gets the best result by LIF term weighting approach mostly. Naive Bayes and Bayes Net both achieve the best performance by LIB*LIF method. SVM and Random Forest get the best result by LIB and LIF respectively.

With the comparison of the classic TF*IDF term weighting approach, the classification performance achieves significant im-

Table 3
Performance impact by different term weighting methods on different classifiers.

Method	Evaluation Metric	RH_1 ($\partial = 0.4$)	RH_2 ($\partial = 0.6$)	RH_3 ($\partial = 0.8$)	RH_4 ($\partial = 1.0$)	SVM	Naive Bayes	Bayes Net	Random Forest
TF*IDF	Precision	0.8340	0.8356	0.8267	0.8339	0.8405	0.7547	0.7526	0.7032
	Recall	0.8207	0.8213	0.8113	0.7953	0.8367	0.7333	0.7420	0.6840
	F1	0.8246	0.8255	0.8151	0.8048	0.8373	0.7315	0.7408	0.6773
LIB	Precision	0.8445	0.8444	**0.8506**	0.8462	**0.8443**	0.4610	0.7946	0.7161
	Recall	0.8307	0.8300	0.8373	0.8147	**0.8407**	0.4333	0.7807	0.6947
	F1	0.8346	0.8343	0.8407	0.8229	**0.8413**	0.3627	0.7801	0.6835
LIF	Precision	**0.8564**	**0.8577**	0.8505	**0.8512**	0.8407	0.7835	0.8401	**0.7502**
	Recall	**0.8420**	**0.8460**	**0.8380**	**0.8173**	0.8380	0.7600	0.8287	**0.7233**
	F1	**0.8460**	**0.8491**	**0.8410**	**0.8263**	0.8384	0.7588	0.8287	**0.7113**
LIB*LIF	Precision	0.8516	0.8487	0.8425	0.8417	0.8375	**0.7840**	**0.8424**	0.7408
	Recall	0.8387	0.8360	0.8287	0.8053	0.8340	**0.7607**	**0.8293**	0.7187
	F1	0.8425	0.8399	0.8320	0.8147	0.8343	**0.7598**	**0.8295**	0.7048

Table 4
t-test result by different term weighting methods (df = 3, RH method).

Paired Comparison	TF*IDF && LIB	TF*IDF && LIF	TF*IDF && LIB*LIF
t-test (Precision)	t(3) = 4.0603 p = 0.0269	t(3) = 15.1132 p = 0.0006	t(3) = 6.3566 p = 0.0079
t-test (Recall)	t(3) = 3.917 p = 0.0296	t(3) = 18.99 p = 0.0003	t(3) = 8.2453 p = 0.0037
t-test (F1)	t(3) = 3.9919 p = 0.0282	t(3) = 21.7467 p = 0.0002	t(3) = 8.2824 p = 0.0037

provement when the new LIT method is in use. The *t*-test result is shown in Table 4 and the p value is lower than 0.05.

5.2.3. Impact of the feature selection method by LIG

We conduct experiments on different feature selection methods, namely IG and LIG. The various term weighting schemes are adopted, including TF*IDF, LIB, LIF and LIB*LIF. The comparison results are shown in Tables 5, 6, 7 and 8 respectively.

For the TF*IDF and LIF term weighting methods which are shown in Tables 5 and 7, the use of LIG feature selection approach improves classification performance in precision, recall and F1 value, except for the Naive Bayes classifier. As shown in Table 8, all of the classifiers based on the LIB*LIF term weighting perform better by the LIG feature selection method than IG feature selection. However, LIG does not work so well for classification when the term weighting method LIB is used. As shown in Table 6, IG performs better than LIG when relaxation hierarchy method (RH_4), SVM and Bayes Net classifiers are implemented.

We analyze the results with *t*-test to compare feature selection methods, IG vs. LIG. The result is shown in Table 9. Except for the LIB term weighting method, the classification performance achieves significant improvement mostly with p value lower than 0.05 when LIG feature selection method is in use.

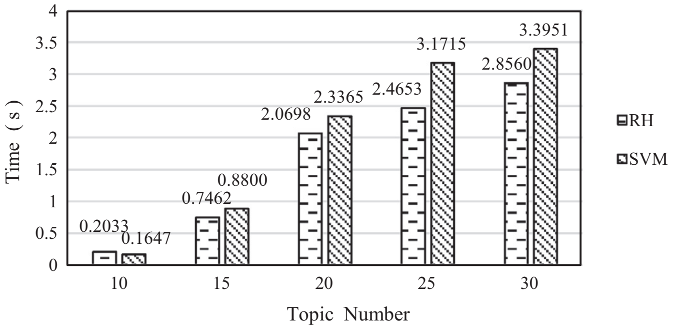


Fig. 4. Classification time comparisons by RH and SVM classifier.

5.2.4. Time performance comparison with SVM

The hierarchical classification method has better time performances especially on larger data sets. We compare average classification time per document between our RH method with ∂ value of 0.6 and a benchmark SVM classifier. The results of classification time vs. the number of topics are shown in Fig. 4. Table 10 shows the efficiency gain(reduction of classification time) of the RH method, compared to SVM.

We find that the efficiency advantage of the RH method over SVM increases when the data size increases. The RH method reduces classification time by more than 10% when the number of topics is greater than 10. Thus, the text hierarchical classification method based on the relaxation strategy improves on classification speed significantly while maintaining both higher precision and recall. This is very important for classification applications on large data sets.

5.3. Discussion

The experimental results presented here show that the relaxation strategy used in the hierarchy classification leads to a signif-

Table 5
Performance impact by different feature selection methods (TF*IDF for term weighting).

Method	Evaluation Metric	RH_1 (∂ = 0.4)	RH_2 (∂ = 0.6)	RH_3 (∂ = 0.8)	RH_4 (∂ = 1.0)	SVM	Naive Bayes	Bayes Net	Random Forest
IG	Precision	0.8450	0.8449	0.8310	0.8465	0.8471	**0.7537**	0.7576	0.6707
	Recall	0.8340	0.8333	0.8181	0.8133	0.8440	**0.7327**	0.7480	0.6320
	F1	0.8375	0.8369	0.8211	0.8216	0.8446	**0.7343**	0.7481	0.6162
LIG	Precision	**0.8510**	**0.8509**	**0.8390**	**0.8535**	**0.8501**	0.7492	**0.7604**	**0.6714**
	Recall	**0.8407**	**0.8387**	**0.8260**	**0.8213**	**0.8467**	0.7267	**0.7500**	**0.6440**
	F1	**0.8436**	**0.8421**	**0.8295**	**0.8294**	**0.8473**	0.7273	**0.7499**	**0.6374**

Table 6
Performance impact by different feature selection methods (LIB for term weighting).

Method	Evaluation Metric	RH_1 (∂ = 0.4)	RH_2 (∂ = 0.6)	RH_3 (∂ = 0.8)	RH_4 (∂ = 1.0)	SVM	Naive Bayes	Bayes Net	Random Forest
IG	Precision	0.8443	0.8436	0.8244	**0.8509**	**0.8521**	0.3200	**0.7750**	0.6440
	Recall	0.8313	0.8300	0.8101	**0.8167**	**0.8467**	0.3967	**0.7687**	0.5700
	F1	0.8350	0.8337	0.8134	**0.8251**	**0.8477**	0.2948	**0.7685**	0.5467
LIG	Precision	**0.8462**	**0.8474**	**0.8297**	0.8483	0.8491	**0.4350**	0.7737	**0.6940**
	Recall	**0.8327**	**0.8333**	**0.8153**	0.8147	0.8440	**0.4000**	0.7667	**0.6587**
	F1	**0.8364**	**0.8372**	**0.8186**	0.8229	0.8449	**0.3041**	0.7666	**0.6465**

Table 7
Performance impact by different feature selection methods (LIF for term weighting).

Method	Evaluation Metric	RH_1 (∂ = 0.4)	RH_2 (∂ = 0.6)	RH_3 (∂ = 0.8)	RH_4 (∂ = 1.0)	SVM	Naive Bayes	Bayes Net	Random Forest
IG	Precision	0.8477	0.8487	0.8481	0.8422	0.8499	**0.7519**	0.7633	0.7010
	Recall	0.834	0.8353	0.8281	0.8100	0.8467	**0.7280**	0.7587	0.6813
	F1	0.8379	0.8390	0.8301	0.8176	0.8476	**0.7301**	0.7557	0.6786
LIG	Precision	**0.8521**	**0.8515**	**0.8520**	**0.8465**	**0.8570**	0.7511	**0.7721**	**0.7028**
	Recall	**0.8387**	**0.8393**	**0.8373**	**0.8180**	**0.8500**	0.7267	**0.7660**	**0.6847**
	F1	**0.8426**	**0.8422**	**0.8398**	**0.8283**	**0.8548**	0.7284	**0.7644**	**0.6936**

Table 8
Performance impact by different feature selection methods (LIB*LIF for term weighting).

Method	Evaluation Metric	RH_1 ($\partial = 0.4$)	RH_2 ($\partial = 0.6$)	RH_3 ($\partial = 0.8$)	RH_4 ($\partial = 1.0$)	SVM	Naive Bayes	Bayes Net	Random Forest
IG	Precision	0.8436	0.8442	0.8419	0.8347	0.8540	0.7500	0.7690	0.6919
	Recall	0.8313	0.8313	0.8285	0.7987	0.8507	0.7287	0.7640	0.6707
	F1	0.8349	0.8350	0.8385	0.8074	0.8516	0.7316	0.7615	0.6653
LIG	Precision	**0.8512**	**0.8502**	**0.8494**	**0.8426**	**0.8588**	**0.7585**	**0.7751**	**0.7075**
	Recall	**0.8387**	**0.8399**	**0.8367**	**0.8067**	**0.8553**	**0.7373**	**0.7693**	**0.6773**
	F1	**0.8422**	**0.8451**	**0.8401**	**0.8185**	**0.8562**	**0.7395**	**0.7677**	**0.6657**

Table 9
t-test result by different feature selection methods.

Paired comparison	IG && LIG (TF*IDF for term weighting)	IG && LIG (LIB for term weighting)	IG && LIG (LIF for term weighting)	IG && LIG (LIB*LIF for term weighting)
t-test (Precision)	t(3) = 14.1 p = 0.0008	t(3) = 1.225 p = 0.3079	t(3) = 10.519 p = 0.0018	t(3) = 17.049 p = 0.0004
t-test (Recall)	t(3) = 11.48 p = 0.0014	t(3) = 1.2864 p = 0.2886	t(3) = 5.1421 p = 0.0143	t(3) = 32.2 p = 0.0001
t-test (F1)	t(3) = 9.279 p = 0.0026	t(3) = 1.2391 p = 0.3034	t(3) = 3.8423 p = 0.0311	t(3) = 3.5288 p = 0.0387

Table 10
Percentage of time reduction by RH method.

Topic number	10	15	20	25	30
Percentage of Time Reduction	−23.44%	15.2%	11.41%	22.27%	15.88%

icant improvement on performance, especially with the combination of the LIT approach for term weighting and feature selection.

By the use of a relaxation strategy, all of the categories are organized as a DAG (Directed Acyclic Graph) structure, which allows each child node to have more than one parent nodes. This approach alleviates the 'blocking' problem effectively. We set different values to the relaxation factor ∂ which controls the hierarchical construction and results in Table 2 show that the system has a degraded classification performance without the relaxation strategy when the ∂ value is set to 1.0. In addition, the hierarchical classification method outperforms other classifiers in terms of classification time. The use of the hierarchical structure leads to significant better efficiency as well as better classification effectiveness.

The Least Information Theory (LIT) is adopted for term weighting and feature selection during the hierarchical classification process. Compared with the TF*IDF classic term weighting method, LIT performs significantly better in terms of precision, recall and F1 score and the *t*-test result is shown in Table 4. The feature selection approach LIG also performs well with different classifiers such as SVM, Bayes Net and Random Forest. As shown in Table 8, all of the classifiers perform better by the LIG feature selection method than IG. The classification performance achieves significant improvement mostly with p value lower than 0.05 which is shown in Table 9.

6. Conclusions

We propose the hierarchical classification approach based on the relaxation strategy which alleviates the impact of the 'blocking' problem. It delays the uncertain category decision until it can be classified definitely, and so the error that has occurred in the upper level will not be transferred to the lower level. We also apply the Least Information Theory in term weighting and documentation representation and it offers a new basic information quantify model by different probability distributions.

The experiments on RCV1 data shows that the text hierarchical classification method based on the relaxation strategy has greater advantage on the time performance over SVM with an increasing data size. And it can also maintain both higher precision and recall simultaneously. Specially, by the use of LIT method for term weighting, the classification performance achieves significant improvement over classic TF*IDF on most classifiers. LIT measures the distance between two probability distributions and establishes a new basic information quantity approach. It performed better than TF*IDF not only in the field of text classification but also other natural language processing application, such as clustering, information retrieval and so on.

In the future, we will optimize and adjust the feature number dynamically at different levels of the hierarchy structure. The node in the upper level contains more categories and it needs ample features for correct classification. On the contrary, the feature number can be reduced for the node in the lower level which contains fewer categories, and it will also improve the time performance effectively. In addition, the proposed approach will be applied to large-scale hierarchical classification task with more class labels and further verify the robustness of the algorithm.

Acknowledgment

This work is supported by the National Science and Technology Support Plan(No. 2013BAH21B02-01), National Nature Science Foundation of China Research Program(61375059, 61672065) and Beijing Natural Science Foundation (No. 4153058).

References

Amati, G., & Rijsbergen, C. J. V. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transsactions on Information System, *20*(4), 357–389.

Aizawa, A. (2000). The feature quantity: An information theoretic perspective of tfidf-like measures. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 104–111). SIGIR '00. New York, NY, USA. ACM.

Bengio, S., Weston, J., & Grangier, D. (2010). Label embedding trees for large multi--class tasks. In *International conference on neural information processing systems* (pp. 163–171). Vancouver, B.C., Canada.

Chen, K., Zhang, Z., Long, J., & Zhang, H. (2016). Turning from TF-IDF to TF-IGM for term weighting in text classification. *Expert Systems with Applications, 66*, 245–260.

Chen, Y. C., Crawford, M. M., & Ghosh, J. (2004). Integrating support vector machines in a hierarchical output space decomposition framework. In *IEEE international symposium on geoscience and remote sensing: 2* (pp. 949–952).

Deepak, A., Kesari, V., & Priyanka, T. (2017). Variable global feature selection scheme for automatic classification of text documents. *Expert Systems with Applications, 81*, 268–281.

Deng, J., Satheesh, S., Berg, A. C., & Li, F. (2011). Fast and balanced: efficient label tree learning for large scale object recognition. *Advances in Neural Information Processing Systems, 24*, 567–575.

Gao, T., & Koller, D. (2011). Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *IEEE international conference on computer vision* (pp. 2072–2079).

Griffin, G., & Perona, P. (2008). Learning and using taxonomies for fast visual categorization. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 1–8). 2008. Alaska, USA.

Gong, X. M., & Ke, W. M. (2015). Term weighting for interactive cluster labeling based on least information gain. *ACM WSDM 2015 workshop on heterogeneous information access (HIA'15). Shanghai, China*.

Jones, K. S. (2004). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 60*, 493–502.

Ke, W. M. (2015). Information-theoretic term weighting schemes for document clustering and classification. *International Journal on Digital Libraries, 16*(2), 145–159.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics, 22*, 79–86.

Lin, J. H. (1991). Divergence measures based on the Shannon entropy. *IEEE Trans on Information Theory, 37*(1), 145–151.

Liu, S., Yi, H. R., Chia, L. T., et al. (2005). Adaptive hierarchical multi-class SVM classifier for texture-based image classification. In *IEEE international conference on multimedia and expo* (pp. 1190–1193).

Liu, T., Liu, S. P., Chen, Z., & Ma, W. Y. (2003). An evaluation on feature selection for text clustering. In *Proceedings of the twentieth international conference on machine learning (ICML 2003)* (pp. 488–495). Washington, DC: AAAI Press.

Marcin, M., & Cordelia, S. (2008). Constructing category hierarchies for visual recognition. In *European conference on computer vision* (pp. 479–491).

Pavlinek, M., & Podgorelec, V. (2017). Text classification method based on self-training and LDA topic models. *Expert Systems with Applications, 80*, 83–93.

Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation, 60*, 503–520.

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends R in Information Retrieval, 3*(4), 333–389.

Viet, H., & Renders, J. M. (2011). Large-Scale hierarchical text classification without labelled data. In *Fourth ACM international conference on web search and data mining* (pp. 685–694). Hong Kong, China.

Wang, Z. F., Wang, Z. H., & Xie, W. J. (2011). Tree-structured bayesian network learning with application to scene classification. *Electronics Letters, 47*(9), 540–541.

Yang, Y. M., & Pedersen, J. O. (1997). A Comparative study on feature selection in text categorization. In *Proceedings of the fourteenth international conference on machine learning (ICML'97)* (pp. 412–420). Nashville, Tennessee, USA.

Zhang, D., Wang, J., & Si, L. (2011). Document clustering with universum. In *Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval* (pp. 873–882). SIGIR'11. New York, NY, USA. ACM.

Zhang, L., Shah, S. K., & Kakadiaris, I. A. (2017). Hierarchical multi-label classification using fully associative ensemble learning. *Pattern Recognition, 70*, 89–103.