Text categorization based on fuzzy classification rules tree

Guo Yuqin[1, 2] Yuan Fang[1] Liu Haibo[1]

([1] College of Mathematics and Computer Science, Hebei University, Baoding 071002, China)

([2] Tianjin Branch of the People's Bank of China, Tianjin 300040, China)

Abstract: To deal with the problem that arises when the conventional fuzzy class-association method applies repetitive scans of the classifier to classify new texts, which has low efficiency, a new approach based on the FCR-tree (fuzzy classification rules tree) for text categorization is proposed. The compactness of the FCR-tree saves significant space in storing a large set of rules when there are many repeated words in the rules. In comparison with classification rules, the fuzzy classification rules contain not only words, but also the fuzzy sets corresponding to the frequencies of words appearing in texts. Therefore, the construction of an FCR-tree and its structure are different from a CR-tree. To debase the difficulty of FCR-tree construction and rules retrieval, more *k*-FCR-trees are built. When classifying a new text, it is not necessary to search the paths of the sub-trees led by those words not appearing in this text, thus reducing the number of traveling rules. Experimental results show that the proposed approach obviously outperforms the conventional method in efficiency.

Key words: text categorization; fuzzy classification association rule; classification rules tree; fuzzy classification rules tree

Automatic text categorization can be defined as assigning class labels to new texts based on the knowledge obtained in a categorization system during the training stage[1]. Several classifiers have been developed and are widely used, such as decision trees, neural networks, *k*-nearest neighbors, support vector machines and association rules. Compared with other methods, categorization based on class-association rules requires a shorter time for training and classifying, and the constructed classifier can be easily understood[2]. Ref. [3] applied fuzzy association rules to classify new texts, but it suffers from some weaknesses regarding class predictions for new texts. For each new text, it should match with all the rules in the classifier to determine the class label, so its efficiency is very low. This paper proposes a fast approach for text categorization. The classification rules are stored in a tree structure in which each node represents a word appearing in the rules. The rules containing the same word can share the prefix sub-tree and thus can save a lot of space in storing rules. When a new text is classified, it is not necessary to traverse the path led by the word not appearing in a particular text. Thus, the number of traveling rules is reduced, which means the classification is accelerated. For the case of classifying multiple texts once, the efficiency is obviously improved.

1 CR-Tree

The CR-tree (classification rules tree) has been introduced in several papers. Ref. [4] applied the CR-tree to store classification rules. In Ref. [4], the mined classification rules are simple and contain little information, so the constructed CR-tree is simple. Ref. [5] presented a text categorization method based on class-association rules with frequencies of words. The classification rules stored in a CR-tree contain not only words, but also the frequencies that words appear in texts. A frequency corresponds to a word and it is real-value, certain and exclusive. When the rules are inserted into a CR-Tree, the frequencies should be reserved with the words. The construction procedure is similar to that in Ref. [4]. Although the frequencies are taken into account, the construction difficulty of the CR-tree is not increased.

2 FCR-Tree

2.1 Construction of FCR-tree

The fuzzy associative classification method proposed in Ref. [3] also considers the frequencies of words, but their disposal is different from those in Ref. [5]. In Ref. [3], the frequencies are expressed by the weights of words. The weight domain is partitioned into three fuzzy sets, and three linguistic terms: *f* (which means the weight is big), *m* (the weight is middle), and *n* (the weight is small) are defined to describe them separately. The form of the fuzzy association rules is shown as follows:

$$\langle w_1, m\rangle, \langle w_2, n\rangle \Rightarrow C_1 \tag{1}$$

$$\langle w_1, f\rangle, \langle w_2, m\rangle \Rightarrow C_1 \tag{2}$$

$$\langle w_1, m\rangle, \langle w_2, n\rangle, \langle w_3, f\rangle \Rightarrow C_2 \tag{3}$$

$$\langle w_4, f\rangle, \langle w_5, n\rangle, \langle w_6, n\rangle \Rightarrow C_3 \tag{4}$$

Different rules can contain the same word, but the weights belonging to the word are different, such as in the first two rules above. To avoid the repetition of the same word, when building a rules tree, we build three branches for each word, one word corresponding to one fuzzy set. For example, inserting the first two rules in the tree, we only need to store w_1 once. But because these fuzzy sets where w_1 corresponds these two rules are different, we need to store w_2 in w_1's branches *m* and *f*, respectively, and then build three branches for w_2. We store the support information of the rules in the branch corresponding to w_2's fuzzy set. That rules tree is called an FCR-tree (fuzzy classification rules tree).

As three fuzzy sets are defined in Ref. [3], we set three

Received 2008-04-15.

Biographies: Guo Yuqin (1981—), female, graduate; Yuan Fang (corresponding author), male, doctor, professor, yuanfang@ hbu. cn.

Foundation items: The National Natural Science Foundation of China (No. 60473045), the Technology Research Project of Hebei Province (No. 05213573), the Research Plan of Education Office of Hebei Province (No. 2004406).

Citation: Guo Yuqin, Yuan Fang, Liu Haibo. Text categorization based on fuzzy classification rules tree[J]. Journal of Southeast University (English Edition), 2008, 24(3): 339 – 342.

branches for every node in an FCR-Tree. If all the rules in a classifier are stored in a tree, a huge tree may be generated, which increases the difficulty of construction and rule retrieval. To solve this problem, we refer to Ref. [6] to construct multiple FCR-trees. The classification rules are divided into subsets by the number of words existing in the rules. A k-FCR-tree is constructed for a subset, where k is the count of words existing in the rules. In Ref. [3], it is necessary that the number of words in the classification rules should be two at least, so we start with a 2-FCR-tree. The construction process can be described as follows:

1) All the words existing in a classifier are sorted according to their frequencies. The order is denoted as S.

2) The words in each rule are sorted according to S.

3) A 2-FCR-tree has a root node, which is set to null.

4) The first words appearing in the rules which only contain two words are registered in the first layer of nodes in the order of S.

5) Three branches are set for each node in the first layer, which correspond to three fuzzy sets f, m, n, respectively. Their values are initialized as 0.

6) For each rule R^2 having two words, the node corresponding to the first word w_1 in R^2 is found in the first layer of the 2-FCR-tree. If the fuzzy set associated with w_1 in R^2 is l_1, the value of l_1 in the 2-FCR-tree should be changed to 1. Then the second word w_2 in R^2 is stored in a second layer of node the in l_1 path.

7) For each node in the second layer, three branches are set up for them in the same way. The class label as well as the support and confidence of R^2 are registered at the last node in the branch corresponding to the fuzzy set associated with w_2. We give an identifier to each rule in the 2-FCR-tree, which is useful for the construction of a 3-FCR-tree.

8) For a 3-FCR-tree, the root node is set to null, and the rules consisting of three words are determined.

9) For each rule R^3 having three words, it is required to judge whether R^3 contains any rule in the 2-FCR-tree. If it contains a rule in the 2-FCR-tree, the identifier of the rule in the 2-FCR-tree can be stored in a first layer of a node in the 3-FCR-tree, which can avoid storing repeated information. And then the remaining word in R^3 is inserted into the tree as a child of this node. The information of R^3 is registered at the last node. When classifying a new text, if a rule in the 2-FCR-tree does not match with it, it is not necessary to travel the path led by the rule's identifier in the 3-FCR-tree, which can reduce the count of traveling rules.

10) If R^3 does not contain any rule in the 2-FCR-tree, it is inserted into the 3-FCR-tree following the construction steps of the 2-FCR-tree. For building a 4-FCR-tree, an identifier is also given to each rule in the 3-FCR-tree.

11) Following the operations above, a k-FCR-tree($k \leqslant$ the largest number of words in the rules) is built in turn.

Fig. 1 shows the structure of a 2-FCR-tree. In general, with the increase of words appearing in the rules, the number of rules will decrease, that means the sharing information is reduced. Besides, the more words the rules contain, the more complex the construction of the FCR-tree is, which increases the difficulty of tree-building and rules retrieval. Therefore, a construction threshold a is set. Before building a k-FCR-tree ($k \geqslant 2$), where the number of the rules containing k words is larger than a is judged. If this condition is satisfied, the rules tree is built, or the rules are stored as they originally appeared.

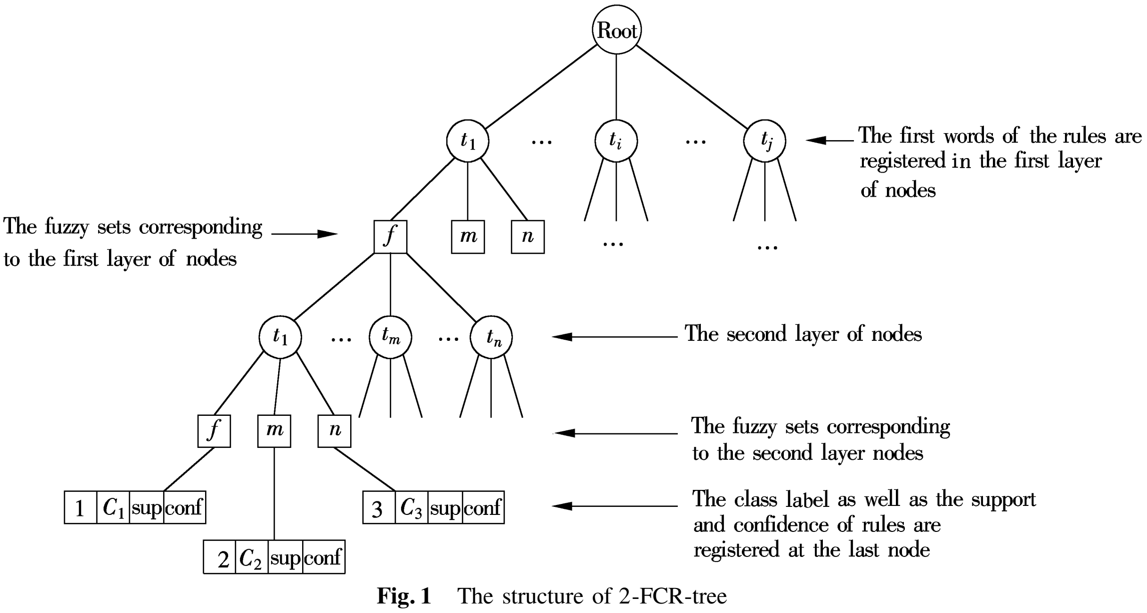
Root
t_1
...
t_i
...
t_j
The first words of the rules are registered in the first layer of nodes
The fuzzy sets corresponding to the first layer of nodes
f
m
n
...
...
t_1
...
t_m
...
t_n
The second layer of nodes
f
m
n
The fuzzy sets corresponding to the second layer nodes
1 | C_1 | sup | conf
3 | C_3 | sup | conf
2 | C_2 | sup | conf
The class label as well as the support and confidence of rules are registered at the last node

Fig. 1 The structure of 2-FCR-tree

2.2 Text categorization based on FCR-tree

After the FCR-tree is completed, it is ready to classify new texts. The classification algorithm based on the FCR-tree can be described as follows:

Algorithm TC_FCRT(t, FCR-tree)//Text categorization based on FCR-tree

Input: k-FCR-trees($k \geqslant 2$); a new text of the form t: $\{\langle w_1, l_1\rangle, \ldots, \langle w_n, l_n\rangle\}$, where w_i is a feature term of t, and l_i is a fuzzy set.

Output: Class label is assigned to the new text t.

Method:

```
1) Travel the 2-FCR-tree first;
2) For each node $w_1$ at the first layer of the 2-FCR-tree
3)   {If $w_1$ appears in $t$
4)     According to the weight of $w_1$ in $t$, search the three branches of $w_1$ to judge whether the value of the fuzzy set corresponding to the weight is 1;
5)       If the value is 1
6)         {Continue to travel down, and judge whether the nodes at the second layer in this path appear in $t$;
7)           If the node $w_2$ at the second layer appears in $t$
8)             {According to the weight of $w_2$ in $t$, search the three branches of $w_2$ to judge whether the value of the fuzzy set corresponding to the weight is 1;
9)               If the value of the fuzzy set is 1   //This rule matches $t$.
10)                Record the identifier as well as the class label, support and confidence of this rule; }}
11)      If the value is 0
12)        Stop searching down, and change to judge the next node at the first layer; }
13) The retrieval over the 2-FCR-tree is completed, and the 3-FCR-tree is traveled in succession;
14) For each node $n_1$ at the first layer of 3-FCR-tree which registers rule's identifier
15)   If $n_1$ is a rule's identifier which has been recorded when the 2-FCR-tree is traveled
16)     Following the operations 5) to 10) to determine the children of $n_1$;
17) For each node $w_1$ at the first layer of the 3-FCR-tree which registers a word
18)   Apply the similar operations as when traveling the 2-FCR-tree to search the path led by $w_1$;
19) Do the above steps to travel the $k$-FCR-tree in turn;
20) For each rule not saved in the FCR-tree
21)   Match it with $t$ directly;
22) Based on the information of the matching rules with $t$ to predict classes for $t$.   //Apply the class prediction method applied in Ref. [3] to assign classes to new texts
```

3 Experimental Results and Performance Analysis

Experimental data comes from a Chinese text database of Fudan University. The database is divided into a training set and a testing set. The categories of the texts contained in the training set and the test set are equal. We only test nine categories carrying out the experiment. The repeated and the incomplete texts are pruned. Tab. 1 shows the numbers of texts of each category in both the training set and the testing set after pruning. We apply the method proposed in Ref. [3] to train a classifier. The obtained classifier consists of 6 562 fuzzy class-association rules.

In order to objectively evaluate our approach, we compare it against the classifier-scan method applied in Ref. [3]. The testing set is divided into six subsets, in which the numbers of texts are almost equal, and two random subsets have no intersection. Experiments are conducted on them using two methods: a classifier-scan method and our proposed method. The classification results of the two methods are identical. We mainly report the execution time, which is shown in Tab. 2.

Tab. 1 Text database

Category	Number	
	Training set	Testing set
Agriculture	842	841
Environment	805	797
Computer	1 007	1 004
Politics	974	973
Space	497	490
Art	509	513
History	466	468
Economy	1 355	1 358
Sports	1 171	1 178
Total	7 626	7 622

Tab. 2 Experimental results

Number of training subsets	Runtime/s		Correct rate/%
	Classifier-scan method	Our method	
1 275	2 209	188	77. 3
1 273	2 273	176	78. 3
1 270	2 401	171	76. 7
1 269	2 234	170	75. 7
1 268	2 295	178	77. 5
1 267	2 166	166	77. 0

From Tab. 2, there are obvious differences in the execution time. Although with our method, the FCR-tree should be built before classifying new texts, the time spent on construction is trivial. In our experiment, it costs 20 s to insert 6 562 rules into the FCR-tree. Relative to the classification time saved by our method, the building time can almost be ignored. Moreover, so long as the training set has no change, the FCR-tree is built only once. As can be seen, the method proposed in this paper is more efficient than the traditional method.

4 Conclusion

With the conventional classification method, the classifier is used to predict class labels directly. For each new text, the classifier should be scanned once, thus the efficiency is very unsatisfactory. This paper proposes a new technique for text categorization. All the rules in the classifier are stored in the FCR-tree. The compactness of the FCR-tree saves significant space in storing a large set of rules when there are many shared words in the rules. In addition, when a new text is classified, it is not necessary to search the path led by the word not appearing in this text in the FCR-tree, thus reducing the number of traveling rules. Because the information contained in fuzzy associative classification rules is more than that contained in general rules, the structure of the FCR-tree is different from that of the CR-tree. To deal with the problem of the complex construction of the FCR-tree, more FCR-trees are built. Experimental results show that our method is feasible and the classification efficiency is obviously improved.

References

[1] Wang Yuanzhen, Qian Tieyun, Feng Xiaonian. Association

rules based automatic Chinese text categorization [J]. *Mini-Micro Systems*, 2005, **26**(8): 1380 – 1383. (in Chinese)

[2] Antonie M L, Zaiane O R. Text document categorization by term association [C]//*Proc of the IEEE International Conference on Data Mining*(*ICDM'*02). Maebashi City, Japan, 2002: 19 – 26.

[3] Yuan Fang, Guo Yuqin, Yang Liu, et al. Chinese text categorization based on fuzzy association rules [C]//*Proc of International Conference on Machine Learning and Cybernetics*. Dalian, China, 2006: 1030 – 1035.

[4] Li Wenmin, Han Jiawei, Pei Jian. CMAR: accurate and efficient classification based on multiple class-association rules [C]//*Proc of the IEEE International Conference on Data Mining*(*ICDM'*01). San Jose, CA, USA, 2001: 369 – 376.

[5] Chen Xiaoyun, Chen Yi, Wang Lei, et al. Text categorization based on classification rules tree by frequent patterns [J]. *Journal of Software*, 2006, **17**(5): 1017 – 1025. (in Chinese)

[6] Song Yuqing, Wang Lijun, Lü Ying, et al. Efficient association rule mining algorithm based on classification tree [J]. *Journal of Jiangsu University*: *Natural Science Edition*, 2006, **27**(1): 51 – 54. (in Chinese)

基于模糊分类规则树的文本分类

郭玉琴[1,2]　袁　方[1]　刘海博[1]

([1] 河北大学数学与计算机学院,保定 071002)

([2] 中国人民银行天津分行,天津 300040)

摘要:针对传统的基于关联规则的文本分类方法在分类文本时需要遍历分类器中的所有规则,分类效率非常低的问题,提出一种基于模糊分类规则树(FCR-tree)的文本分类方法. 分类器中的规则以树的形式存储,由于树型结构避免了重复结点的存储,节省了存储空间. 模糊分类关联规则与一般分类规则相比,不仅包含了词条信息,还包含了词条出现频度对应的模糊集,所以 FCR-tree 的构建过程及树的结构不同于一般规则树 CR-tree. 为降低构建及遍历 FCR-tree 的难度,采用了构造多棵 *k*-FCR-tree 的方法. 在搜索规则树时,如果结点中的词条没在待分类文本中出现,则不需要再搜索该结点引导的子树,大大减少了需要匹配的规则的数量. 实验表明该方法是可行的,与遍历分类器的分类方法相比,分类效率有了明显提高.

关键词:文本分类;模糊分类关联规则;分类规则树;模糊分类规则树

中图分类号:TP393