Contents lists available at ScienceDirect

Information Processing and Management

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

Mining fuzzy frequent itemsets for hierarchical document clustering

Chun-Ling Chen [a], Frank S.C. Tseng [b,*], Tyne Liang [a]

[a] Department of Computer Science, National Chiao Tung University, HsinChu 300, Taiwan, ROC
[b] Dept. of Information Management, National Kaohsiung 1st University of Science and Technology, YenChao, Kaohsiung 824, Taiwan, ROC

A R T I C L E   I N F O

Article history:
Received 20 October 2008
Received in revised form 28 September 2009
Accepted 29 September 2009
Available online 31 October 2009

Keywords:
Fuzzy association rule mining
Text mining
Hierarchical document clustering
Frequent itemsets

A B S T R A C T

As text documents are explosively increasing in the Internet, the process of hierarchical document clustering has been proven to be useful for grouping similar documents for versatile applications. However, most document clustering methods still suffer from challenges in dealing with the problems of high dimensionality, scalability, accuracy, and meaningful cluster labels. In this paper, we will present an effective Fuzzy Frequent Itemset-Based Hierarchical Clustering (F^2IHC) approach, which uses fuzzy association rule mining algorithm to improve the clustering accuracy of Frequent Itemset-Based Hierarchical Clustering (FIHC) method. In our approach, the key terms will be extracted from the document set, and each document is pre-processed into the designated representation for the following mining process. Then, a fuzzy association rule mining algorithm for text is employed to discover a set of highly-related fuzzy frequent itemsets, which contain key terms to be regarded as the labels of the candidate clusters. Finally, these documents will be clustered into a hierarchical cluster tree by referring to these candidate clusters. We have conducted experiments to evaluate the performance based on Classic4, Hitech, Re0, Reuters, and Wap datasets. The experimental results show that our approach not only absolutely retains the merits of FIHC, but also improves the accuracy quality of FIHC.

Crown Copyright © 2009 Published by Elsevier Ltd. All rights reserved.

1. Introduction

In order to browse and organize documents smoothly, hierarchical clustering techniques have been proposed to cluster a collection of documents into a hierarchical tree structure. Despite that, there still exist several challenges for hierarchical document clustering, such as high dimensionality, scalability, accuracy, and meaningful cluster labels (Beil, Ester, & Xu, 2002; Fung, Wang, & Ester, 2002, 2003).

As text mining is much more complex than data mining because text data are inherently unstructured and fuzzy (Tan, 1999), some studies (Delgado, MartÃn-Bautista, SÃanchez, & Vila, 2002; Feldman & Dagan, 1995; Lin et al., 1998) applied the technique of association rule mining in document management. For example, Feldman and Dagan (1995) have presented a Knowledge Discovery in Text (KDT) system, which used the simplest information extraction approach to get interesting information and knowledge from unstructured text collections. Lin et al. (1998) proposed a method, namely Mining Term Association, to acquire the semantic relations between terms when applying to documents. Moreover, Delgado et al. (2002) think that the association rule mining is the first data mining technique employed in mining text collections. It is very interesting since many applications related to text processing involve associations and co-occurrence between terms. These works mainly focused on analyzing the co-occurrence terms for document management.

* Corresponding author. Present address: 1, University Road, YenChao, Kaoshiung County 824, Taiwan, ROC. Tel.: +886 7 6011000x4113; fax: +886 7 6011042.
E-mail addresses: chunling@cs.nctu.edu.tw (C.-L. Chen), imfrank@ccms.nkfust.edu.tw (F.S.C. Tseng), tliang@cs.nctu.edu.tw (T. Liang).

0306-4573/$ - see front matter Crown Copyright © 2009 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2009.09.009

Furthermore, Fung et al. (2002) proposed a novel method, namely Frequent Itemset-based Hierarchical Clustering (FIHC), to produce a hierarchical topic tree for document clustering. This method offers some merits to resolve the challenges, such as dimensionality reduction, the number of clusters as an optional input parameter, and easy browsing with meaningful cluster labels. They employed the *tf-idf* (Term Frequency-Inverse Document Frequency) method (Salton, 1971) to replace the actual term frequency of a term by the weighted frequency. However, the main limitation of *tf-idf* method is that long documents tend to have higher weights than short ones. This is because it considers only the weighted frequency of the terms in a document, but neglects the length of the document. This disadvantage will affect the accuracy of the clustering task, when the mining algorithm cannot obtain appropriate topic labels for the derived clusters.

In this paper, we will propose an approach which stems from prior studies (Hong, Lin, & Wang, 2003; Kaya & Alhajj, 2006; Martín-Bautista, Sánchez, Chamorro-Martínez, Serrano, & Vila, 2004), by integrating fuzzy set concepts (Zadeh, 1965) and the association rule mining to find interesting fuzzy association rules from given transactions. The fuzzy association rule mining is a good method choice because it is easily understandable and realistic for integrating linguistic terms with fuzzy sets.

Compared with the *tf-idf* weighting used in FIHC, we intend to focus on using the fuzzy association rule mining based on term frequency to find the association relationships between terms for clustering documents. Since some important terms that express the topics of a document may rarely appear in the document collection, if we use the association rule mining instead, then only the terms which frequently occur in the document collection can be retrieved, which implies that the important sparse terms might be obscured in the process of document clustering. By applying the fuzzy association rule mining, we can discover interesting connections between fuzzy frequent itemsets. For example, $(t_1 \cdot Low \rightarrow t_2 \cdot High)$ or even $(t_1 \cdot Low \rightarrow t_2 \cdot Low)$ are rules that can be found to show the association relationships between the frequencies of important terms in the document collection.

In order to flexibly apply the frequent itemset-based technique for more applications in document clustering, we extend our previous study (Chen, Tseng, Liang, 2008) and further propose an effective Fuzzy Frequent Itemset-Based Hierarchical Clustering (F^2IHC) approach based on the fuzzy association rule mining to ameliorate the accuracy quality of FIHC. In contrast with our previous study, we explain our approach in more details and conduct experiments to evaluate more datasets.

Our approach can be distinguished into the following stages:

1. In the first stage, the key terms will be extracted from the document set, and each document is pre-processed into the designated representation for the following mining process. In this stage, a hybrid feature selection method will be used to effectively reduce the unimportant terms for each document.
2. In the second stage, to discover a set of relevant fuzzy frequent itemsets efficiently, we will propose a fuzzy association rule mining algorithm for text. In this algorithm, we revise the method devised by Hong et al. (2003) by regarding a document as a transaction, and those term frequency values in a document as the quantitative values (i.e., the number of purchased items in a transaction). A frequent itemset, as defined by Fung et al. (2003), is a set of words that occur together in some minimum fraction of documents in a cluster. By employing pre-defined membership functions, our algorithm calculates three fuzzy values, i.e., *Low*, *Mid*, and *High* regions, for each term based on its frequency to discriminate the degree of importance of the term within a document in the mining process. The derived fuzzy frequent itemsets contain key terms to be regarded as the labels of candidate clusters.
3. In the final stage, the documents will be clustered into a hierarchical cluster tree based on these candidate clusters. The cluster tree will be built in a top-down fashion to recursively select the parent clusters at level $k-1$ for dividing the documents into its suitable children clusters at level k. Notice that the clusters generated by our algorithm are crisp partitions for assigning a document to exactly one cluster.

In summary, our approach has the following advantages:

1. It provides a frequent itemset-based clustering algorithm for the analysis of a document set to generate a flexible hierarchical document cluster tree, which can be easily integrated into a document management system for providing flexible browsing and retrieving of various applications.
2. It shows how the fuzzy association rule mining can be applied more accurately to document clustering. It extends a fuzzy data representation used in data mining by Hong et al. (2003) to text mining to provide a subtler partitioning of a dataset.
3. By conducting experimental results to evaluate the datasets of Classic4, Hitech, Re0, Reuters, and Wap, it has been proven that our approach not only absolutely retains the merits of FIHC in reducing the high dimensionality of text, efficiency for these datasets, and the meaningful labels of the discovered clusters, but also improves the accuracy quality of FIHC.

The subsequent sections of this paper are organized as follows: In Section 2, we briefly review related literature on document clustering methods. Section 3 presents our approach in three stages, together with an illustrative example. Experimental results are presented and analyzed in Section 4. Finally, we conclude and propose some future directions in Section 5.

2. Document clustering methods

In general, major clustering algorithms are divided into partitioning methods and hierarchical methods. For document clustering, partitioning methods exclusively partition the set of documents into a number of clusters by moving documents from one cluster to another. A common partitioning method, namely *k*-means method, has been used in (Bellot & El-Beze, 1999; Hu, Zhou, Guan, & Hu, 2008; Iliopoulos, Enright, & Ouzounis, 2001).

The main propose of hierarchical document clustering is to build a hierarchical tree of clusters whose leaf nodes represent the subset of a document collection. Moreover, this method can be further classified into agglomerative and divisive approaches, which work in a bottom-up and top-down fashion, respectively. An agglomerative clustering iteratively merges two most similar clusters until a terminative condition is satisfied. On the other hand, a divisive method starts with one cluster, which consists of all documents, and recursively splits one cluster into smaller sub-clusters until some termination criterion is fulfilled.

Besides, frequent itemsets have also been applied to document clustering extensively. For example, Beil et al. (2002) proposed the Hierarchical Frequent Term-based Clustering (HFTC) method by using frequent itemsets and minimizing the overlap of clusters in terms of shared documents. However, the experiments of Fung et al. (2003) showed that HFTC is not scalable. For a scalable algorithm, Fung et al. proposed a novel FIHC algorithm by using frequent itemsets derived from the association rule mining to construct a hierarchical topic tree for clusters. They also proved that using frequent itemsets for document clustering can reduce the dimension of a vector space effectively. Therefore, this approach not only reduces dimensionality, but also offers efficient processing of high volume data, supports ease of browsing, and provides meaningful cluster labels.

Steinbach, Karypis, and Kumar (2000) have compared the performance of some influential clustering algorithms, and the results indicated that UPGMA and Bisecting *k*-means are the most accurate clustering algorithms. Therefore, we will compare the performance of our algorithm with that of FIHC, UPGMA, and Bisecting *k*-means algorithms in term of accuracy in Section 4.

In the following, we will present our approach, which stems from the frequent itemset-based clustering technique, and extends the algorithm proposed by Hong et al. (2003) to text processing, to find suitable fuzzy frequent itemsets for constructing a hierarchical cluster tree.

3. The framework of our approach

There are three stages in our framework as shown in Fig. 1. We explain them as follows:

1. *Document pre-processing*. By document pre-processing, the frequency of each term within a document is counted.
2. *Candidate clusters extraction*. Use our fuzzy association rule mining algorithm to find fuzzy frequent itemsets, which are then used to form the candidate clusters.
3. *The cluster tree construction*. Build the Document-Cluster Matrix (DCM) for assigning each document to a fitting cluster. Then, a pruned hierarchical cluster tree will be built.

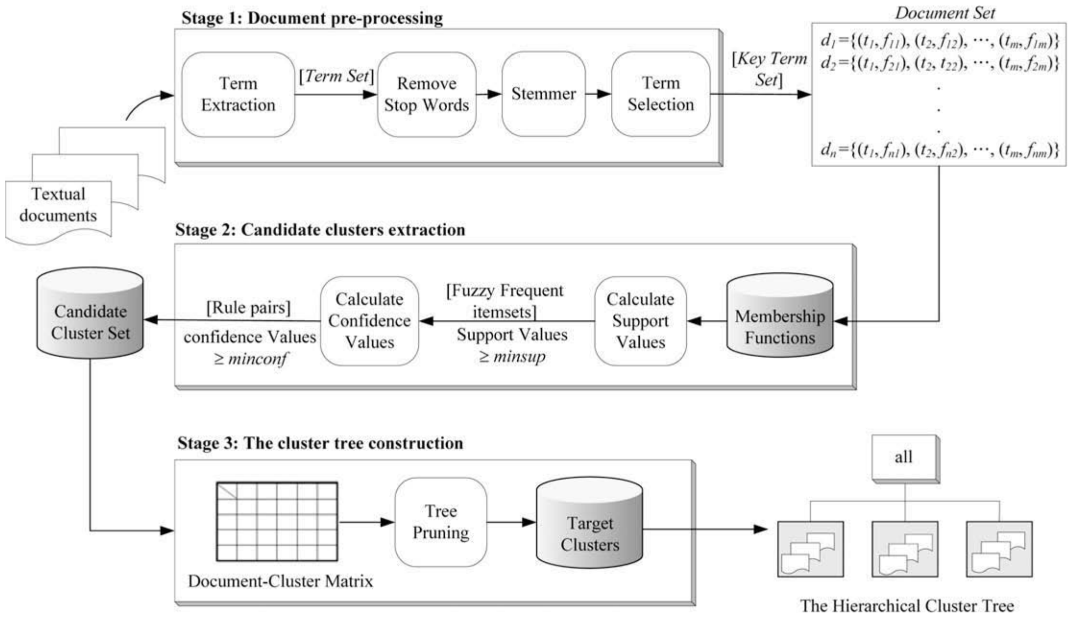

Stage 1: Document pre-processing
Term Extraction
[Term Set]
Remove Stop Words
Stemmer
Term Selection
[Key Term Set]
Document Set
$d_1=\{(t_1,f_{11}),(t_2,f_{12}),\cdots,(t_m,f_{1m})\}$
$d_2=\{(t_1,f_{21}),(t_2,t_{22}),\cdots,(t_m,f_{2m})\}$
$d_n=\{(t_1,f_{n1}),(t_2,f_{n2}),\cdots,(t_m,f_{nm})\}$
Textual documents
Stage 2: Candidate clusters extraction
Candidate Cluster Set
[Rule pairs]
confidence Values $\geq minconf$
Calculate Confidence Values
[Fuzzy Frequent itemsets]
Support Values $\geq minsup$
Calculate Support Values
Membership Functions
Stage 3: The cluster tree construction
Document-Cluster Matrix
Tree Pruning
Target Clusters
all
The Hierarchical Cluster Tree

Fig. 1. The framework of our approach.

3.1. Stage 1: document pre-processing

This stage describes the required transformation processes of documents to obtain the desired representation of documents. As there are thousands of words in a document set, the purpose of this stage is to reduce dimensionality for high clustering accuracy. Several methods, such as itemset pruning (Beil et al., 2002), feature clustering or co-clustering (Mandhani, Joshi, & Kummamuru, 2003), feature selection technique (Shihab, 2004), and matrix factorization (Shahnaz, Berry, Pauca, & Plemmons, 2006; Xu & Gong, 2004), have been applied to reduce dimensionality. To solve this problem, we have to find the terms that are significant and important to represent the content of each document. Hence, we must remove the terms that are not meaningful and discriminative to increase the clustering accuracy and maintain the computing cost small. We describe the details of the pre-processing in the following:

1. *Divide the sentences into terms.*
2. *Remove the stop words.* We use a stop word list[1] that contains words to be excluded. The list is applied to remove the terms that have general meaning but do not discriminate for topics.
3. *Conduct word stemming*: Use the developed stemming algorithms, such as Porter (1980), to reduce a word to its stem or root form.
4. *Term selection*. The terms with selection metric weights all higher than pre-specified thresholds will be selected as key terms. In our approach, three feature selection methods, *tf-idf*, *tf-df*, and *tf*^{2}, are used to select representative terms for each document, and these feature selection methods are defined as follows:
   (1) *tf-idf* (term frequent × inverse document frequency): It is denoted as $tfidf_{ij}$ and used for the measure of the importance of term t_j within document d_i. For preventing a bias for longer documents, the weighted frequency of each term is usually normalized by the total frequencies of all terms in document d_i, and is defined as follows:

$$tfidf_{ij} = \frac{f_{ij}}{\sum_{j=1}^{m} f_{ij}} \times log\left(\frac{|D|}{|\{d_i|t_j \in d_i, d_i \in D\}|}\right) \tag{3.1}$$

   where f_{ij} is the frequency of term t_j in document d_i, and the denominator is the total frequencies of all terms in document d_i. $|D|$ is the total number of documents in the document set D, and $|\{d_i \mid t_j \in d_i, d_i \in D\}|$ is the number of documents containing term t_j.
   (2) *tf-df* (term frequent × document frequency): It is represented by $tfdf_{ij}$ and evaluated by (3.2) for the value calculated by dividing the term frequency (*TF*) by the document frequency (*DF*), where *TF* is the number of times a term t_j appears in a document d_i divided by the total frequencies of all terms in d_i, and *DF* is used to determine the number of documents containing term t_j divided by the total number of documents in the document set D:

$$tfdf_{ij} = TF/DF, \qquad \text{where } TF = \frac{f_{ij}}{\sum_{j=1}^{m} f_{ij}}, \quad \text{and} \quad DF = \frac{|\{d_i|t_j \in d_i, d_i \in D\}|}{|D|} \tag{3.2}$$

   (3) *tf*^{2}: It is the multiplication of $tfidf_{ij}$ and $tfdf_{ij}$, and we denote it as tf_{ij}^2:

$$tf_{ij}^2 = tfidf_{ij} * tfdf_{ij} \tag{3.3}$$

After these weights of each term in each document have been calculated, those which have weights all higher than pre-specified thresholds are retained. Subsequently, these retained terms form a set of key terms for the document set *D*, and we formally define them as follows.

Definition 3.1. A *document*, denoted $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \ldots, (t_j, f_{ij}), \ldots, (t_m, f_{im})\}$, is a logical unit of text, characterized by a set of key terms t_j together with their corresponding frequency f_{ij}.

Definition 3.2. A *document set*, denoted $D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$, also called a *document collection*, is a set of documents, where n is the total number of documents in D.

Definition 3.3. The *term set* of a document set $D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$, denoted $T_D = \{t_1, t_2, \ldots, t_j, \ldots, t_s\}$, is the set of terms appeared in D, where s is the total number of terms.

Definition 3.4. The key term set of a document set $D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$, denoted $KD = \{t_1, t_2, \ldots, t_j, \ldots, t_m\}$, is a subset of the term set T_D, including only meaningful key terms, which do not appear in a well-defined stop word list, and satisfy the pre-defined minimum *tf-idf* threshold α, the minimum *tf-df* threshold β and the minimum *tf*^{2} threshold γ.

Based on these definitions, the representation of a document can be derived by Algorithm 3.1 shown in Fig. 2. Let us consider, for example, a document set $D = \{d_1, d_2, \ldots, d_{10}\}$ containing 10 documents. By Algorithm 3.1, we might obtain the de-

[1] It contains a list of 571 stop words that was developed by the SMART project.

Algorithm 3.1: The document pre-processing algorithm for deriving the representation of all documents in a document set *D*.

```
Input: 1. A document set $D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$.
       2. A well-defined stop word list.
       3. The minimum tf-idf threshold $\alpha$.
       4. The minimum tf-df threshold $\beta$.
       5. The minimum tf$^2$ threshold $\gamma$.
Output: The key term set of $D$, $K_D$.
        The formal representation of all documents in $D$ as defined in Definition 3.1.
Method:
Step 1. Extract the term set $T_D = \{t_1, t_2, \ldots, t_j, \ldots, t_s\}$.
Step 2. Remove all stop words from $T_D$.
Step 3. Apply Word Stemming for $T_D$.
Step 4. For each $d_i \in D$ do
          For each $t_j \in T_D$ do
            (1) Evaluate its $tfidf_{ij}$, $tfdf_{ij}$, and $tf^2$ weights.
            (2) Retain the term if $tfidf_{ij} \geq \alpha$, $tfdf_{ij} \geq \beta$, and $tf^2_{ij} \geq \gamma$.
Step 5. Obtain the key term set $K_D$ based on the previous steps.
Step 6. For each $d_i \in D$ do
          For each $t_j \in K_D$ do
            (1) count its frequency in $d_i$ to obtain $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \ldots, (t_j, f_{ij}), \ldots, (t_m, f_{im})\}$.
```

Fig. 2. A detailed illustration of Algorithm 3.1.

Table 1
Document set.

Docs ID	Key term set					
	Stock	Record	Profit	Medical	Treatment	Health
d_1	2	1	1	0	0	0
d_2	1	1	0	0	0	0
d_3	1	0	2	0	0	0
d_4	0	0	0	3	0	2
d_5	0	0	0	11	1	1
d_6	0	1	0	4	0	0
d_7	0	0	0	8	1	2
d_8	3	0	1	0	0	0
d_9	0	1	0	3	0	0
d_{10}	0	0	0	8	2	1

rived representation of D and its key term set K_D = {stock, record, profit, medical, treatment, health} as shown in Table 1. Notice that we use a tabular representation, where each entry denotes the frequency of a key term (the column heading) in a document d_i (the row heading), to make our presentation more concise. This representation scheme will be employed in the following to illustrate our approach.

3.2. Stage 2: candidate clusters extraction

The objective of this stage is to take a document set D, a set of pre-defined membership functions, the minimum support value θ, and the minimum confidence value λ as input, and to output a set of candidate clusters. To achieve this goal, we modified the algorithm proposed by Hong et al. (2003) to capture the relationships among different key terms of the document set. Since each discovered fuzzy frequent itemset has an associated fuzzy count value, it can be regarded as the degree of importance that the itemset contributes to the document set.

In the following, we will define the membership functions, present our algorithm, and finally explain our approach by an illustrative example.

3.2.1. The membership functions

The membership functions are used to convert each term frequency into a fuzzy set. Therefore, we define the *t-f* (term frequency) fuzzy set used in this paper as follows.

Definition 3.5. A *t-f fuzzy set* of document d_i is a pair (F_{ij}, w^r_{ij}), where F_{ij} is a set and equals to $\{w^{Low}_{ij}(f_{ij})/t_j \cdot Low,\ w^{Mid}_{ij}(f_{ij})/t_j \cdot Mid,\ w^{High}_{ij}(f_{ij})/t_j \cdot High\}$, $w^r_{ij} : F \to [0, 2]$, and r can be *Low*, *Mid*, or *High*. The notation $t_j \cdot r$ is called a fuzzy region of t_j. For each term pair (t_j, f_{ij}) of document d_i, $w^r_{ij}(f_{ij})$ is the grade of membership of t_j in d_i with *Low*, *Mid*, and *High* membership functions defined by Formulas (3.4)–(3.6), respectively.

$$w^{Low}_{ij}(f_{ij}) = \begin{cases} 0, & f_{ij} = 0 \\ 1 + f_{ij}/a_1, & 0 < f_{ij} < a_1 \\ 2, & f_{ij} = a_1 \\ 1 + a_2 - f_{ij}/a_2 - a_1, & a_1 < f_{ij} < a_2 \\ 1, & f_{ij} \geq a_2 \end{cases}, \quad a_1 = min(f_{ij}), \quad a_2 = avg(f_{ij}) \tag{3.4}$$

$$w^{Mid}_{ij}(f_{ij}) = \begin{cases} 0, & f_{ij} = 0 \\ 1, & f_{ij} = a_1 \\ 1 + f_{ij} - a_1/a_2 - a_1, & a_1 < f_{ij} < a_2 \\ 2, & f_{ij} = a_2 \\ 1 + a_3 - f_{ij}/a_3 - a_2, & a_2 < f_{ij} < a_3 \\ 1, & f_{ij} = a_3 \end{cases}, \quad a_1 = min(f_{ij}), \quad a_2 = avg(f_{ij}), \quad a_3 = max(f_{ij}) \tag{3.5}$$

$$w^{High}_{ij}(f_{ij}) = \begin{cases} 0, & f_{ij} = 0 \\ 1, & f_{ij} \leq a_1 \\ 1 + f_{ij}/a_2 - a_1, & a_1 < f_{ij} < a_2 \\ 2, & f_{ij} = a_2 \end{cases}, \quad a_1 = avg(f_{ij}), \quad a_2 = max(f_{ij}) \tag{3.6}$$

In Formulas (3.4)–(3.6), $min(f_{ij})$ is the minimum frequency of terms in D, $max(f_{ij})$ is the maximum frequency of terms in D, and $avg(f_{ij}) = \left\lceil \frac{\sum_{i=1}^{n} f_{ij}}{|K|} \right\rceil$, where $f_{ij} \neq min(f_{ij})$ or $max(f_{ij})$, and $|K|$ is the number of summed key terms. For example, based on the document set in Table 1, the derived membership functions are shown in Fig. 3.

3.2.2. The fuzzy association rule mining algorithm for text

To describe our fuzzy association rule mining algorithm shown, we need the following definitions.

Definition 3.6. For a document set D, a *candidate cluster* $\tilde{c} = (\tilde{D}_c, \tau)$ is a two-tuple, where $\tilde{D}_c$ is a subset of the document set D, such that it includes those documents which contain all the key terms in $\tau = \{t_1, t_2, \ldots, t_q\} \subseteq K_D$, $q \geqslant 1$, where K_D is the key term set of D and q is the number of key terms included in τ. In fact, τ is a fuzzy frequent itemset for describing $\tilde{c}$. To illustrate, $\tilde{c}$ can also be denoted as $\tilde{c}^q_{(t_1,t_2,\ldots,t_q)}$ or $\tilde{c}^q_{(\tau)}$, and will be used interchangeably hereafter. For instance, in Table 1, the *candidate cluster* $\tilde{c}^1_{(\text{stock})} = (\{d_1, d_2, d_3, d_8\}, \{\text{stock}\})$, as the term "stock" appeared in these documents.

Definition 3.7. The *candidate cluster set* of a document set D, denoted $\tilde{C}_D = \{\tilde{c}^1_1, \ldots, \tilde{c}^2_{l-1}, \tilde{c}^q_l, \ldots, \tilde{c}^q_k\}$, is a set of candidate clusters, where k is the total number of candidate clusters. The candidate cluster set $\tilde{C}_D$ for a document set D can be generated by Algorithm 3.2 shown in Fig. 4.

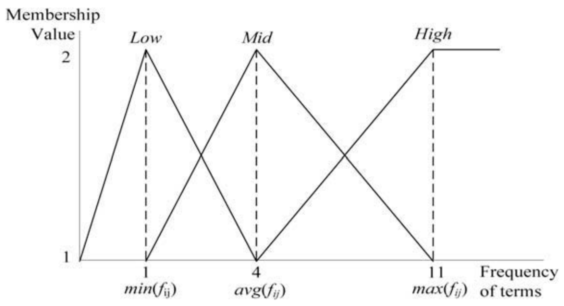


Fig. 3. The pre-defined membership functions of the example.

Algorithm 3.2. The fuzzy association rule mining algorithm for finding fuzzy frequent itemsets and regarded them as a candidate cluster set for output.

```
Input: 1. A document set $D = \{d_1, d_2, \ldots, d_n\}$, where $d_i = \{(t_1, f_{i1}), (t_2, f_{i2}), \ldots, (t_j, f_{ij}), \ldots, (t_m, f_{im})\}$;
       2. A set of membership functions (as defined in Section 3.2.1);
       3. The minimum support value $\theta$;
       4. The minimum confidence value $\lambda$.
Output: Generate a candidate cluster set $\tilde{C}_D$.
Method:
Step 1. For each key term, using the given membership functions to transform each term frequency $f_{ij}$ into a fuzzy set
        $F_{ij} = w_{ij}^{Low}/t_j.Low + w_{ij}^{Mid}/t_j.Mid + w_{ij}^{High}/t_j.High.$
Step 2. For $D$, calculate the scalar cardinalities of three fuzzy regions for each key term $t_j \in K_D$. That is,
        $count_j^{Low} = \sum_{i=1}^{n} w_{ij}^{Low}$, $count_j^{Mid} = \sum_{i=1}^{n} w_{ij}^{Mid}$, and $count_j^{High} = \sum_{i=1}^{n} w_{ij}^{High}$
Step 3. Find the region of each key term with maximum count. That is,
        $max\text{-}count_j = max(count_j^{Low}, count_j^{Mid}, count_j^{High})$, for $j = 1$ to $m$, where $m$ is the total number of key terms in $K_D$.
        Let $max\text{-}R_j$ be the region with $max\text{-}count_j$ for a key term $t_j$, which will be used to represent the fuzzy weight of $t_j$
        in the following steps.
Step 4. Find fuzzy frequent 1-itemsets $L_1$.
        This step checks the support value of a key term $t_j$, denoted $support(t_j) = \frac{max\text{-}count_j}{|D|}$, $1 \le j \le m$. If it is larger
        than or equal to $\theta$, where $|D|$ is the number of documents in $D$ and $max\text{-}count_j$ is the count value of a region
        $max\text{-}R_j$, then the fuzzy frequent itemset will be put into the large 1-itemset $L_1$. That is,
        $L_1 = \{max\text{-}R_j \mid support(t_j) \ge \theta, 1 \le j \le m\}$.
Step 5. Generate the candidate set $C_2$ from $L_1$.
Step 6. Find fuzzy frequent 2-itemsets $L_2$.
        For each candidate 2-itemset $\tau$ with key terms $(t_1, t_2)$ in $C_2$, there are three sub-steps to be performed:
        6-1. Generate the fuzzy value of $\tau$, $w_{i\tau} = min(w_{i1}^{max\text{-}R_1}, w_{i2}^{max\text{-}R_2})$, in each document $d_i$, where $w_{ij}^{max\text{-}R_j}$ is the fuzzy
             value of the maximum region of a key term $t_j$ in $d_i$.
        6-2. Calculate the scalar cardinality of $\tau$ in $D$ as $count_\tau = \sum_{i=1}^{n} w_{i\tau}$.
        6-3. Put $\tau$ in $L_2$ if $support(\tau) = \frac{count_\tau}{|D|}$ is larger than or equal to $\theta$.
Step 7. Check whether $L_2$ is null. If $L_2$ is null, then exit the algorithm; otherwise, do the next step.
Step 8. Set $q = 2$, where $q$ represents the number of key terms stored in the current fuzzy frequent $q$-itemsets.
Step 9. Generate the candidate set $C_{q+1}$ from $L_q$.
        This step is similar to the a priori algorithm (de Campos & Moral, 1993). First, our algorithm also joins $L_q$ and $L_q$
        to assume that $q$ - 1 key terms in the two itemsets are the same and the other one is different. Then, $C_{q+1}$ itemsets
        that have all their sub-$q$-itemsets in $L_q$ are generated.
Step 10. Find fuzzy frequent $(q + 1)$-itemsets $L_{q+1}$.
        For each candidate $(q + 1)$-itemsets $\tau$ with key terms $(t_1, t_2, \ldots, t_{q+1})$ in $C_{q+1}$, there are three sub-steps to be
        achieved:
        10-1. Generate the fuzzy value of $\tau$, $w_{i\tau} = min\{w_{ij}^{max\text{-}R_j} \mid j = 1, 2, \ldots, q+1\}$, in each document $d_i$, where is the fuzzy
              value of the maximum region of a key term $t_j$ in $d_i$.
        10-2. Calculate the scalar cardinality of $\tau$ in $D$ as $count_\tau = \sum_{i=1}^{n} w_{i\tau}$.
        10-3. Put $\tau$ in $L_{q+1}$, if $support(\tau) = \frac{count_\tau}{|D|}$ is larger than or equal to $\theta$.
Step 11. Check whether $L_{q+1}$ is null or not.
        If $L_{q+1}$ is null, then exit the algorithm; otherwise, set $q = q + 1$ and repeat Steps 9-11.
Step 12. Construct the association rules.
        For all the fuzzy frequent $q$-itemsets $\tau$ containing key terms $(t_1, t_2, \ldots, t_q)$, where $q \ge 2$. There are sub-steps to be
        accomplished:
        12-1. Form all possible association rules. That is,
              $\tau_1 \wedge \ldots \wedge \tau_{k-1} \wedge \tau_{k+1} \wedge \ldots \wedge \tau_q \to \tau_k$, $k = 1$ to $q$.
        12-2. Calculate the confidence values of all association rules using the formula:
              $\frac{\sum_{i=1}^{n} w_{i\tau}}{\sum_{i=1}^{n} (w_{i1} \wedge \ldots \wedge w_{ik-1}, w_{ik+1} \wedge \ldots \wedge w_{iq})}$
Step 13. Hold those rules which the confidence values of the rule pair are all larger than or equal to $\lambda$.
Step 14. Output the fuzzy frequent 1-itemsets and fuzzy frequent $q$-itemsets derived in Step 13 to form the candidate
        cluster set $\tilde{C}_D$.
```

Fig. 4. A detailed illustration of Algorithm 3.2.

3.2.3. An illustrative example

Consider using the document set D in Table 1, the membership functions defined in Fig. 3, the minimum support value 40%, and the minimum confidence value 60% as inputs. The fuzzy frequent itemsets discovery procedure is illustrated in Fig. 5.

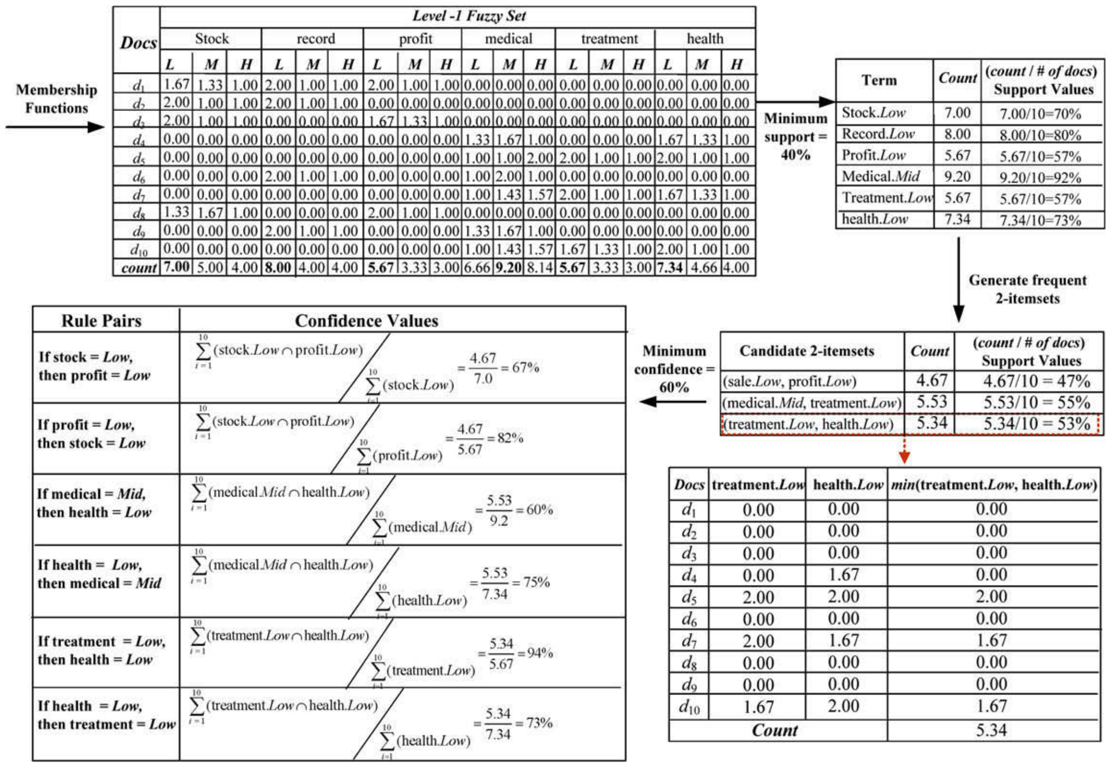
In the proposed algorithm, we estimate the strength of association among key terms in the document set by using confidence values. There is useful information when the co-occurring keywords have been shown. This is because highly co-occurring terms are used together. Thus, our algorithm computes the confidence values of a rule pair to check the strong association of key terms $(t_1, t_2, \dots, t_q)$ of the fuzzy frequent q-itemsets. Take the candidate cluster $\tilde{c}^2_{(\text{stock,profit})}$ as an example. Since its confidence value of the rule pair "If stock = *Low*, then profit = *Low*" and "If profit = *Low*, then stock = *Low*" are both larger than the minimum confidence value 60%, $\tilde{c}^2_{(\text{stock,profit})}$ is put in the candidate cluster set $\tilde{C}_D$. Finally, the candidate cluster set $\tilde{C}_D = \left\{\tilde{c}^1_{(\text{stock})}, \tilde{c}^1_{(\text{record})}, \tilde{c}^1_{(\text{profit})}, \tilde{c}^1_{(\text{medical})}, \tilde{c}^1_{(\text{treatment})}, \tilde{c}^1_{(\text{health})}, \tilde{c}^2_{(\text{stock,profit})}, \tilde{c}^2_{(\text{medical,health})}, \tilde{c}^2_{(\text{treatment,health})}\right\}$ will be output.

3.3. Stage 3: the cluster tree construction

The candidate cluster set generated by the previous steps can be viewed as a set of topics with their corresponding subtopics in the document set. In this stage, we first construct the Document-Term Matrix (DTM) and the Term-Cluster Matrix (TCM) to derive the Document-Cluster matrix (DCM) for assigning each document to a fitting cluster, such that each c^q_i contains a subset of documents. For the documents in each c^q_i, the intra-cluster similarity is minimized and the inter-clusters similarity is maximized. We call each c^q_i a *target cluster* in the following. Based on the assignment result, we will find the set of target clusters $C_D = \{c^1_1, c^1_2, \dots, c^q_i, \dots, c^q_f\}$, and then use these target clusters to form a hierarchical tree for the document set D. To avoid the constructed cluster tree including too many clusters, the methods described in Section 3.3.3 will be used to prune unnecessary clusters.

3.3.1. Building the Document-Cluster Matrix (DCM)

First, consider each candidate cluster $\tilde{c}^q_{(\tau)} = \tilde{c}^q_{(t_1,t_2,\dots,t_q)}$ with fuzzy frequent itemset τ. τ will be regarded as a reference point for generating a target cluster. Then, to represent the degree of importance of document d_i in a candidate cluster $\tilde{c}^q_l$, an $n \times k$ Document-Cluster Matrix will be constructed to calculate the similarity of terms in d_i and $\tilde{c}^q_l$. To achieve this goal, we have to define two matrixes, namely Document-Term Matrix and Term-Cluster Matrix, as follows.



Docs	Level -1 Fuzzy Set																	
	Stock			record			profit			medical			treatment			health		
	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
d_1	1.67	1.33	1.00	2.00	1.00	1.00	2.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d_2	2.00	1.00	1.00	2.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d_3	2.00	1.00	1.00	0.00	0.00	0.00	1.67	1.33	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d_4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.33	1.67	1.00	0.00	0.00	0.00	1.67	1.33	1.00
d_5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	2.00	2.00	1.00	1.00	2.00	1.00	1.00
d_6	0.00	0.00	0.00	2.00	1.00	1.00	0.00	0.00	0.00	1.00	2.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
d_7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.43	1.57	2.00	1.00	1.00	1.67	1.33	1.00
d_8	1.33	1.67	1.00	0.00	0.00	0.00	2.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
d_9	0.00	0.00	0.00	2.00	1.00	1.00	0.00	0.00	0.00	1.33	1.67	1.00	0.00	0.00	0.00	0.00	0.00	0.00
d_{10}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.43	1.57	1.67	1.33	1.00	2.00	1.00	1.00
count	**7.00**	5.00	4.00	**8.00**	4.00	4.00	**5.67**	3.33	3.00	6.66	**9.20**	8.14	**5.67**	3.33	3.00	**7.34**	4.66	4.00

Term	*Count*	*(count / # of docs)* Support Values
Stock.*Low*	7.00	7.00/10=70%
Record.*Low*	8.00	8.00/10=80%
Profit.*Low*	5.67	5.67/10=57%
Medical.*Mid*	9.20	9.20/10=92%
Treatment.*Low*	5.67	5.67/10=57%
health.*Low*	7.34	7.34/10=73%

Candidate 2-itemsets	*Count*	*(count / # of docs)* Support Values
(sale.*Low*, profit.*Low*)	4.67	4.67/10 = 47%
(medical.*Mid*, treatment.*Low*)	5.53	5.53/10 = 55%
(treatment.*Low*, health.*Low*)	5.34	5.34/10 = 53%

Docs	treatment.*Low*	health.*Low*	*min*(treatment.*Low*, health.*Low*)
d_1	0.00	0.00	0.00
d_2	0.00	0.00	0.00
d_3	0.00	0.00	0.00
d_4	0.00	1.67	0.00
d_5	2.00	2.00	2.00
d_6	0.00	0.00	0.00
d_7	2.00	1.67	1.67
d_8	0.00	0.00	0.00
d_9	0.00	0.00	0.00
d_{10}	1.67	2.00	1.67
Count			5.34

Rule Pairs	Confidence Values
If stock = *Low*, then profit = *Low*	$\sum_{i=1}^{10}(\text{stock}.Low \cap \text{profit}.Low) \big/ \sum_{i=1}^{10}(\text{stock}.Low) = \frac{4.67}{7.0} = 67\%$
If profit = *Low*, then stock = *Low*	$\sum_{i=1}^{10}(\text{stock}.Low \cap \text{profit}.Low) \big/ \sum_{i=1}^{10}(\text{profit}.Low) = \frac{4.67}{5.67} = 82\%$
If medical = *Mid*, then health = *Low*	$\sum_{i=1}^{10}(\text{medical}.Mid \cap \text{health}.Low) \big/ \sum_{i=1}^{10}(\text{medical}.Mid) = \frac{5.53}{9.2} = 60\%$
If health = *Low*, then medical = *Mid*	$\sum_{i=1}^{10}(\text{medical}.Mid \cap \text{health}.Low) \big/ \sum_{i=1}^{10}(\text{health}.Low) = \frac{5.53}{7.34} = 75\%$
If treatment = *Low*, then health = *Low*	$\sum_{i=1}^{10}(\text{treatment}.Low \cap \text{health}.Low) \big/ \sum_{i=1}^{10}(\text{treatment}.Low) = \frac{5.34}{5.67} = 94\%$
If health = *Low*, then treatment = *Low*	$\sum_{i=1}^{10}(\text{treatment}.Low \cap \text{health}.Low) \big/ \sum_{i=1}^{10}(\text{health}.Low) = \frac{5.34}{7.34} = 73\%$

Fig. 5. The process of Algorithm 3.2 of this example.

$$W = \begin{array}{c|cccc} & t_1 & t_2 & \cdots & t_p \\ \hline d_1 & w_{11}^{\max-R_j} & w_{12}^{\max-R_j} & \cdots & w_{1p}^{\max-R_j} \\ d_2 & w_{21}^{\max-R_j} & w_{22}^{\max-R_j} & \cdots & w_{2p}^{\max-R_j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_n & w_{n1}^{\max-R_j} & w_{n2}^{\max-R_j} & \cdots & w_{np}^{\max-R_j} \end{array}_{n\times p}$$

Fig. 6. A formal illustration of Document-Term Matrix.

$$G = \begin{array}{c|cccccc} & \tilde{c}_1^1 & \cdots & \tilde{c}_{l-1}^1 & \tilde{c}_l^q & \cdots & \tilde{c}_k^q \\ \hline t_1 & g_{11}^{\max-R_j} & \cdots & g_{1l-1}^{\max-R_j} & g_{1l}^{\max-R_j} & \cdots & g_{1k}^{\max-R_j} \\ t_2 & g_{21}^{\max-R_j} & \cdots & g_{2l-1}^{\max-R_j} & g_{2l}^{\max-R_j} & \cdots & g_{2k}^{\max-R_j} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ t_p & g_{p1}^{\max-R_j} & \cdots & g_{pl-1}^{\max-R_j} & g_{pl}^{\max-R_j} & \cdots & g_{pk}^{\max-R_j} \end{array}_{p\times k}$$

Fig. 7. A formal illustration of Term-Cluster Matrix.

Definition 3.8. A *Document-Term Matrix* (*DTM*), denoted $W = \left[w_{ij}^{\max-R_j}\right]$, for a document set D, is an $n \times p$ matrix, such that $w_{ij}^{\max-R_j}$ is the weight (fuzzy value) of term t_j in document d_i and $t_j \in L_1$. A formal illustration of DTM can be found in Fig. 6.

Definition 3.9. A *Term-Cluster Matrix* (*TCM*), denoted $G = \left[g_{jl}^{\max-R_j}\right]$, for a document set D of n documents, is an $p \times k$ matrix, such that for $1 \leqslant j \leqslant p$, $1 \leqslant l \leqslant k$, and

$$g_{jl}^{\max-R_j} = \frac{score(\tilde{c}_l^q)}{\sum_{i=1}^{n} w_{ij}^{\max-R_j}}, \quad \text{where } score(\tilde{c}_l^q) = \begin{cases} \sum\limits_{d_i \in \tilde{c}_l^1, t_j \in L_1} w_{ij}^{\max-R_j} & \text{if } q = 1, \\ \dfrac{\sum_{d_i \in \tilde{c}_l^q, t_j \in L_1} w_{ij}^{\max-R_j}}{\lambda} & \text{else,} \end{cases} \tag{3.7}$$

In Formula (3.7), $w_{ij}^{\max-R_j}$ is the weight (fuzzy value) of term t_j in document $d_i \in \tilde{c}_l^q$ and λ is the minimum confidence value.

Each $g_{jl}^{\max-R_j}$ of TCM represents the degree of importance of key term t_j in a candidate cluster $\tilde{c}_{(\tau)}^q$ by referring to those documents including τ. To reduce the dimension, only key terms present in L_1 were applied in TCM. A formal illustration of TCM can be found in Fig. 7.

Finally, based on the previous two definitions, we can define the Document-Cluster Matrix (DCM) of a document set D as follows.

Definition 3.10. A *Document-Cluster Matrix* (*DCM*) for a document set D of n documents is the inner product of its DTM and TCM. It is an $n \times k$ matrix, and can be defined as $V = [v_{il}]$, where

$$v_{il} = row_i(W) \cdot col_l(G) = \left[w_{i1}^{\max-R_j}\, w_{i2}^{\max-R_j} \cdots w_{ip}^{\max-R_j}\right] \begin{bmatrix} g_{1l}^{\max-R_j} \\ g_{2l}^{\max-R_j} \\ \vdots \\ g_{pl}^{\max-R_j} \end{bmatrix} = \sum_{p=1}^{p} w_{ip} g_{pl}, \quad 1 \le i \le n \quad \text{and} \quad 1 \le l \le k$$

A formal illustration of DCM can be found in Fig. 8.

3.3.2. Building the hierarchical cluster tree

Based on the obtained DCM, each document can be assigned to only one target cluster by using the following rules.

Rule 1. Each element v_{il} of the DCM matrix represents the degree of importance of document d_i in a candidate cluster $\tilde{c}_l^1$. For each document d_i (the row i of DCM), if there exists only one maximum v_{il} in $\{v_{i1}, v_{i2}, \ldots, v_{iy}\} \in \tilde{c}_{(\tau)}^1$ (the column 1 to y of DCM), where $1 \leqslant y \leqslant k$, then d_i will be assigned to a target cluster c_l^1; otherwise, apply Rule 2.

$$V = \begin{array}{c} \\ d_1 \\ d_2 \\ \vdots \\ d_n \end{array}\!\!\begin{array}{c} \begin{array}{cccccc} \tilde{c}^1_{11} \cdots & \tilde{c}^2_{1l-1} & \tilde{c}^q_{1l} & \cdots & \tilde{c}^q_{1k} \end{array} \\ \left[\begin{array}{cccccc} v_{11} \cdots & v_{1l-1} & v_{1l} & \cdots & v_{1k} \\ v_{21} \cdots & v_{2l-1} & v_{2l} & \cdots & v_{2k} \\ \vdots \ddots & \vdots & \vdots & \ddots & \vdots \\ v_{n1} \cdots & v_{nl-1} & v_{nl} & \cdots & v_{nk} \end{array}\right] \\ n \times k \end{array} = \begin{array}{c|cccc} & t_1 & t_2 & \cdots & t_p \\ \hline d_1 & & & & \\ d_2 & w^{\max-R_j}_{21} & w^{\max-R_j}_{22} & \cdots & w^{\max-R_j}_{2p} \\ \vdots & & & & \\ d_n & & & & \end{array}_{n \times p} \cdot \begin{array}{c|cccc} & \tilde{c}^1_1 & \tilde{c}^1_2 & \cdots & \tilde{c}^q_k \\ \hline t_1 & & g^{\max-R_j}_{12} & & \\ t_2 & & g^{\max-R_j}_{22} & & \\ \vdots & & \vdots & & \\ t_p & & g^{\max-R_j}_{p2} & & \end{array}_{p \times k}$$

Fig. 8. A formal illustration of Document-Cluster Matrix.

$$c^1_l = \{d_i | v_{il} = \max\{v_{i1}, v_{i2}, \ldots, v_{iy}\} \in \tilde{c}^1_{(\tau)}, \quad \text{where } 1 \le y \le k\} \tag{3.8}$$

Rule 2. If a document d_i has the same maximum values $\{v_{i1}, v_{i2}, \ldots, v_{iy}\} \in \tilde{c}^1_{(\tau)}$ from more than one of the candidate clusters $\{\tilde{c}^1_1, \tilde{c}^1_2, \ldots, \tilde{c}^1_y\}$, then d_i will be assigned to a target cluster c^1_l, such that its fuzzy frequent itemset τ has the highest count value. Notice that when $q = 1$, the count value is *max-count$_j$* (refer to the Step 3 in Algorithm 3.2).

After assigning each document to the best fitting cluster, the resulting tree can be formed as a foundation for pruning and a natural structure for browsing. The cluster tree built by F^{2}IHC algorithm has the following eight features:

1. The cluster tree is built in a top-down fashion, which is different from the cluster tree obtained in a bottom-up fashion by FIHC.
2. Each child target cluster has exactly one parent target cluster.
3. The topic of a parent target cluster is more general than the topic of its children target clusters. The nodes become more and more specialized as they get closer to the leaf nodes.
4. A parent target cluster and its children target clusters are "similar" to a certain degree.
5. Each target cluster employs one fuzzy frequent q-itemset τ as its cluster label.
6. The root node of the cluster tree appears at level 0, and is tagged with the cluster label "all".
7. Each target cluster with its fuzzy frequent q-itemset appears in the level q of the tree.
8. The depth of the cluster tree is the same as the maximum size of fuzzy frequent itemsets.

3.3.3. Tree pruning

When a low minimum support value and a low minimum confidence value are used, the target cluster tree would become broad and deep. The documents with the same topic may be spread to several small target clusters, which would cause low document clustering accuracy. In order to generate a natural hierarchical cluster tree for higher document clustering accuracy and for easy browsing, one tree pruning method is used for merging similar target clusters at level 1. This method employs the following definition to compute the inter-cluster similarity between two target clusters.

Definition 3.11. The *inter-cluster similarity* between two target clusters c^1_x and $c^1_y, c^1_x \neq c^1_y$, is defined by Formula (3.9):

$$Inter\_Sim(c^1_x, c^1_y) = \frac{\sum^n_{d_i \in c^1_x, c^1_y} v_{ix} \times v_{iy}}{\sqrt{\sum^n_{d_i \in c^1_x} (v_{ix})^2 \times \sum^n_{d_i \in c^1_y} (v_{iy})^2}} \tag{3.9}$$

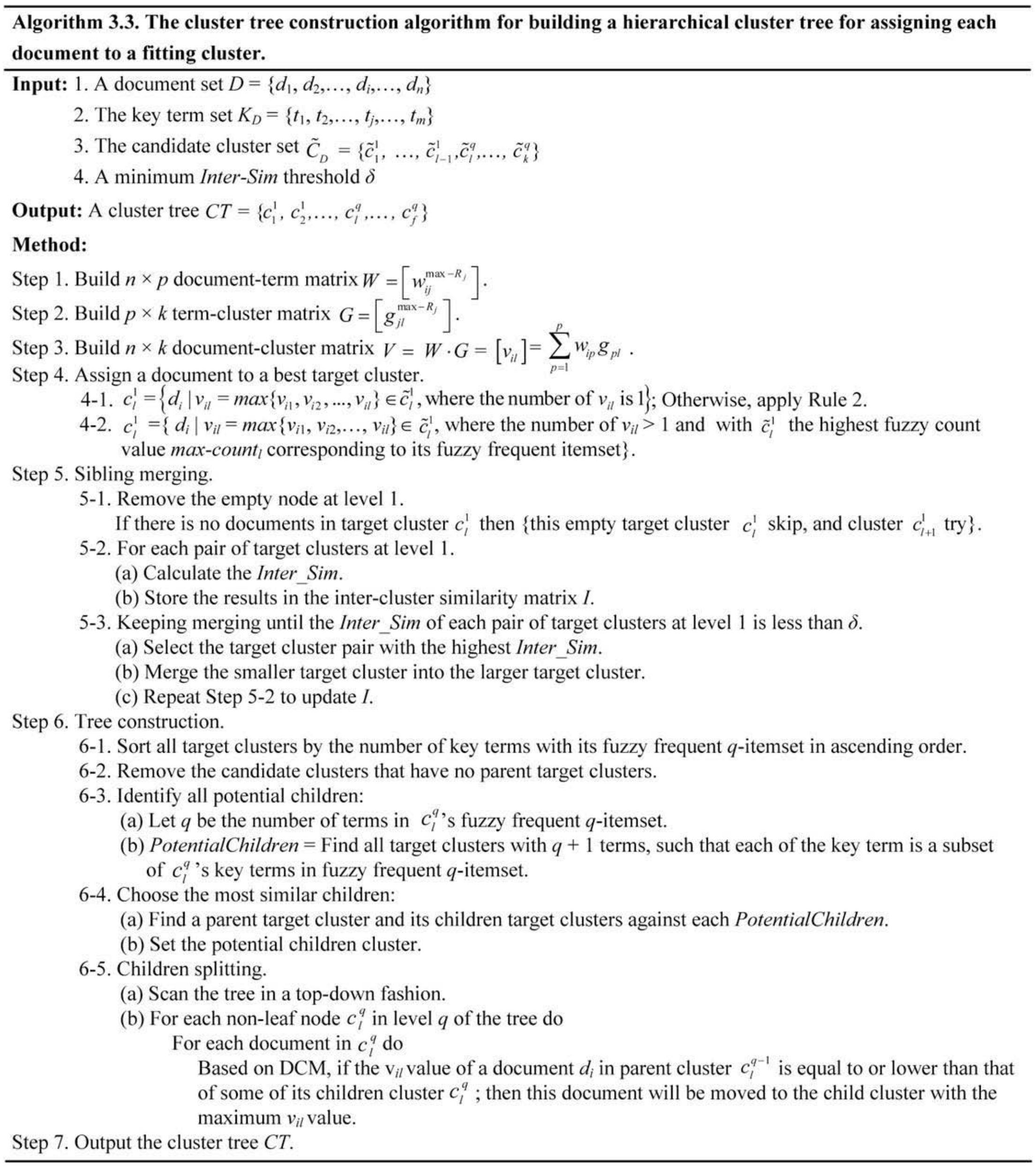
where v_{ix} and v_{iy} stand for two entries, such that $d_i \in c^1_x, d_i \in c^1_y$, in DCM, respectively; The range of *Sim* is [0, 1]. If the *Inter-Sim* value is close to 1, then both clusters are regarded nearly the same. In the following, the minimum *Inter-Sim* will be used as a threshold δ to decide whether two target clusters should be merged.

The objective of sibling merging is to shrink a tree by merging similar target clusters at level 1 for attaining high document clustering accuracy. Each pair of target clusters at level 1 of a tree is calculated by using the inter-cluster similarity measure. The target cluster pair with the highest *Inter-Sim* value must keep merging until the *Inter-Sim* value of all target clusters at level 1 becomes smaller than the minimum *Inter-Sim* threshold δ.

Algorithm 3.3 shown in Fig. 9 is used to assign each document to the best fitting cluster, and finally builds a cluster tree for output.

3.3.4. An illustrative example

For example, consider the document set in Table 1. The key term set K_D = {stock, record, profit, medical, treatment, health} and the candidate cluster set $\tilde{C}_D = \left\{\tilde{c}^1_{(stock)}, \tilde{c}^1_{(record)}, \tilde{c}^1_{(profit)}, \tilde{c}^1_{(medical)}, \tilde{c}^1_{(treatment)}, \tilde{c}^1_{(health)}, \tilde{c}^2_{(stock,profit)}, \tilde{c}^2_{(medical,health)}, \tilde{c}^2_{(treatment,health)}\right\}$



Algorithm 3.3. The cluster tree construction algorithm for building a hierarchical cluster tree for assigning each document to a fitting cluster.

```
Input: 1. A document set $D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$
       2. The key term set $K_D = \{t_1, t_2, \ldots, t_j, \ldots, t_m\}$
       3. The candidate cluster set $\tilde{C}_D = \{\tilde{c}_1^1, \ldots, \tilde{c}_{l-1}^1, \tilde{c}_l^q, \ldots, \tilde{c}_k^q\}$
       4. A minimum Inter-Sim threshold $\delta$
Output: A cluster tree $CT = \{c_1^1, c_2^1, \ldots, c_l^q, \ldots, c_f^q\}$
Method:
Step 1. Build $n \times p$ document-term matrix $W = \left[ w_{ij}^{\max - R_j} \right]$.
Step 2. Build $p \times k$ term-cluster matrix $G = \left[ g_{jl}^{\max - R_j} \right]$.
Step 3. Build $n \times k$ document-cluster matrix $V = W \cdot G = [v_{il}] = \sum_{p=1}^{p} w_{ip} g_{pl}$ .
Step 4. Assign a document to a best target cluster.
    4-1. $c_l^1 = \{ d_i \mid v_{il} = max\{v_{i1}, v_{i2}, \ldots, v_{il}\} \in \tilde{c}_l^1$, where the number of $v_{il}$ is 1$\}$; Otherwise, apply Rule 2.
    4-2. $c_l^1 = \{ d_i \mid v_{il} = max\{v_{i1}, v_{i2}, \ldots, v_{il}\} \in \tilde{c}_l^1$, where the number of $v_{il} > 1$ and with $\tilde{c}_l^1$ the highest fuzzy count
         value $max\text{-}count_l$ corresponding to its fuzzy frequent itemset$\}$.
Step 5. Sibling merging.
    5-1. Remove the empty node at level 1.
         If there is no documents in target cluster $c_l^1$ then {this empty target cluster $c_l^1$ skip, and cluster $c_{l+1}^1$ try}.
    5-2. For each pair of target clusters at level 1.
         (a) Calculate the Inter_Sim.
         (b) Store the results in the inter-cluster similarity matrix I.
    5-3. Keeping merging until the Inter_Sim of each pair of target clusters at level 1 is less than $\delta$.
         (a) Select the target cluster pair with the highest Inter_Sim.
         (b) Merge the smaller target cluster into the larger target cluster.
         (c) Repeat Step 5-2 to update I.
Step 6. Tree construction.
    6-1. Sort all target clusters by the number of key terms with its fuzzy frequent q-itemset in ascending order.
    6-2. Remove the candidate clusters that have no parent target clusters.
    6-3. Identify all potential children:
         (a) Let q be the number of terms in $c_l^q$'s fuzzy frequent q-itemset.
         (b) PotentialChildren = Find all target clusters with q + 1 terms, such that each of the key term is a subset
             of $c_l^q$'s key terms in fuzzy frequent q-itemset.
    6-4. Choose the most similar children:
         (a) Find a parent target cluster and its children target clusters against each PotentialChildren.
         (b) Set the potential children cluster.
    6-5. Children splitting.
         (a) Scan the tree in a top-down fashion.
         (b) For each non-leaf node $c_l^q$ in level q of the tree do
                 For each document in $c_l^q$ do
                     Based on DCM, if the $v_{il}$ value of a document $d_i$ in parent cluster $c_l^{q-1}$ is equal to or lower than that
                     of some of its children cluster $c_l^q$ ; then this document will be moved to the child cluster with the
                     maximum $v_{il}$ value.
Step 7. Output the cluster tree CT.
```

Fig. 9. A detailed illustration of Algorithm 3.3.

were already generated in Sections 3.1 and 3.2.3, respectively. Now, suppose the minimum *Inter-Sim* value is 0.6. The proposed cluster tree construction algorithm proceeds as follows:

Step 1. Build 10×6 Document-Term Matrix W in Table 2.
Step 2. Build 6×9 Term-Cluster Matrix G in Table 3.
Step 3. Build 10×9 Document-Cluster Matrix V in Table 4.
Step 4. Assign each document to its best target cluster.

$$c^1_{(\text{stock})} = \{d_1, d_3, d_8\} \quad c^1_{(\text{record})} = \{d_2\} \quad c^1_{(\text{medical})} = \{d_4, d_5, d_6, d_7, d_9, d_{10}\}$$
$$c^1_{(\text{profit})} = \{\} \quad c^1_{(\text{treatment})} = \{\} \quad c^1_{(\text{health})} = \{\}$$

Table 2
The DTM of this example.

Documents/key terms	Stock · *Low*	Record · *Low*	Profit · *Low*	Medical · *Mid*	Treatment · *Low*	Health · *Low*
d_1	1.67	2.00	2.00	0.00	0.00	0.00
d_2	2.00	2.00	0.00	0.00	0.00	0.00
d_3	2.00	0.00	1.67	0.00	0.00	0.00
d_4	0.00	0.00	0.00	1.67	0.00	1.67
d_5	0.00	0.00	0.00	1.00	2.00	2.00
d_6	0.00	2.00	0.00	2.00	0.00	0.00
d_7	0.00	0.00	0.00	1.43	2.00	1.67
d_8	1.33	0.00	2.00	0.00	0.00	0.00
d_9	0.00	2.00	0.00	1.67	0.00	0.00
d_{10}	0.00	0.00	0.00	1.43	1.67	2.00

Table 3
The TCM of this example.

Key terms/clusters	$\tilde{c}^1_{(\text{stock})}$	$\tilde{c}^1_{(\text{record})}$	$\tilde{c}^1_{(\text{profit})}$	$\tilde{c}^1_{(\text{medical})}$	$\tilde{c}^1_{(\text{treatment})}$	$\tilde{c}^1_{(\text{health})}$	$\tilde{c}^2_{(\text{stock, profit})}$	$\tilde{c}^2_{(\text{medical, health})}$	$\tilde{c}^2_{(\text{treatment, health})}$
Stock · *Low*	1.00	0.52	0.71	0.00	0.00	0.00	1.19	0.00	0.00
Record · *Low*	0.50	1.00	0.25	0.50	0.00	0.00	0.42	0.00	0.00
Profit · *Low*	1.00	0.35	1.00	0.00	0.00	0.00	1.67	0.00	0.00
Medical · *Mid*	0.00	0.40	0.00	1.00	0.42	0.60	0.00	1.00	0.70
Treatment · *Low*	0.00	0.00	0.00	1.00	1.00	1.00	0.00	1.67	1.67
Health · *Low*	0.00	0.00	0.00	1.00	0.77	1.00	0.00	1.67	1.29

Table 4
The DCM of this example.

Documents/clusters	$\tilde{c}^1_{(\text{stock})}$	$\tilde{c}^1_{(\text{record})}$	$\tilde{c}^1_{(\text{profit})}$	$\tilde{c}^1_{(\text{medical})}$	$\tilde{c}^1_{(\text{treatment})}$	$\tilde{c}^1_{(\text{health})}$	$\tilde{c}^2_{(\text{stock, profit})}$	$\tilde{c}^2_{(\text{medical, health})}$	$\tilde{c}^2_{(\text{treatment, health})}$
d_1	**4.67**	3.58	3.69	1.00	0.00	0.00	6.15	0.00	0.00
d_2	3.00	**3.05**	1.93	1.00	0.00	0.00	3.21	0.00	0.00
d_3	**3.67**	1.64	3.10	0.00	0.00	0.00	5.16	0.00	0.00
d_4	0.00	0.67	0.00	**3.34**	1.99	2.67	0.00	4.46	3.32
d_5	0.00	0.40	0.00	**5.00**	3.96	4.60	0.00	7.67	6.61
d_6	1.00	2.80	0.50	**3.00**	0.84	1.20	0.83	2.00	1.40
d_7	0.00	0.57	0.00	**5.10**	3.89	4.53	0.00	7.55	6.48
d_8	**3.33**	1.40	2.95	0.00	0.00	0.00	4.92	0.00	0.00
d_9	1.00	**2.67**	0.50	**2.67**	0.70	1.00	0.83	1.67	1.17
d_{10}	0.00	0.57	0.00	**5.10**	3.81	4.53	0.00	7.55	6.36

Numbers appeared in boldface mean the largest values of each row of $\tilde{c}^1_\tau$.

Step 5. Sibling merging.

1. Remove the empty node $\{c^1_{(profit)}, c^1_{(treatment)}, c^1_{(health)}\}$.
2. The Inter_Sim values of all pairs of target clusters are calculated in Table 5.
3. Keep merging until the Inter_Sim of all pairs of target clusters are lower than the minimum *Inter-Sim* value 0.6.
   (a) Based on the above result, the cluster pair $(c^1_{(\text{stock})}, c^1_{(\text{record})})$ has the highest Inter_Sim value.
   (b) Since the number of documents of $c^1_{(record)}$ is less than $c^1_{(stock)}$, the document in $c^1_{(record)}$ is merged into $c^1_{(stock)}$. Thus, $c^1_{(stock)} = \{d_1, d_2, d_3, d_8\}$.
   (c) Update the inter-cluster similarity matrix. We omit the details here due to space limitation.

Step 6. Tree construction.

1. Sort all target clusters based on the number of key terms, we obtain $\{c^1_{(stock)}, c^1_{(medical)}, c^2_{(stock,profit)}, c^2_{(medical,health)}, c^2_{(treatment,health)}\}$.
2. Remove the target clusters and it has no parent clusters to produce the result $\{c^2_{(treatment,health)}\}$.
3. Identify all potential children.
   (a) The number of terms in $c^2_{(stock,profit)}$ and $c^2_{(medical,health)}$ are both 2.
   (b) The *PotentialChildren* of $c^1_{(stock)}$ is $c^2_{(stock,profit)}$ and the *PotentialChildren* of $c^1_{(medical)}$ is $c^2_{(medical,health)}$.
4. The target clusters $c^2_{(stock,profit)}$ and $c^2_{(medical,health)}$ are set as the child cluster of $c^1_{(stock)}$ and $c^1_{(medical)}$, respectively.

Table 5
The *Inter-Sim* values of all target clusters.

Cluster pairs (c_x, c_y)	*Inter_Sim*
($c_{(\text{stock})}$, $c_{(\text{record})}$)	0.94
($c_{(\text{stock})}$, $c_{(\text{medical})}$)	0.14
($c_{(\text{record})}$, $c_{(\text{medical})}$)	0.34

Table 6
The compare results between the parent cluster $\tilde{c}^1_{(\text{medical})}$ and its child cluster $\tilde{c}^2_{(\text{medical, health})}$.

Documents/clusters	$\tilde{c}^1_{(\text{medical})}$	$\tilde{c}^2_{(\text{medical, health})}$	Whether the document is divided into the child cluster
d_4	3.34	**4.46**	Yes
d_5	5.00	**7.67**	Yes
d_6	**3.00**	2.00	No
d_7	5.10	**7.55**	Yes
d_9	**2.67**	1.67	No
d_{10}	5.10	**7.55**	Yes

Numbers appeared in boldface mean the largest values of each row.

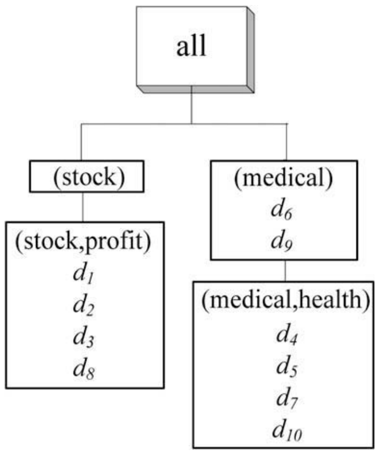


Fig. 10. The derived hierarchical cluster tree.

5. Children splitting.
   (a) Here, we take the documents in the parent cluster $c^1_{(medical)}$ for example.
   (b) Based on DCM, we compare the value v_{il} of each document in the parent cluster $c^1_{(medical)}$ and its child cluster $c^2_{(medical,health)}$ to decide whether the document is divided into the child cluster. The result is shown in Table 6.

Step 7. Finally, the derived cluster tree *CT* can be shown in Fig. 10.

4. Experimental results

In this section, we experimentally evaluate the performance of the proposed algorithm by comparing with that of the FIHC method. We make use of the FIHC 1.0 tool[2] to generate the results of FIHC. The produced results are then fetched into the same evaluation program to ensure a fair comparison. All the experiments have been performed on a P4 3.2 GHz Windows XP machine with 1 GB memory.

4.1. Datasets

We used the five standard datasets employed by the FIHC experiments. These datasets are widely adopted as standard benchmarks for the text categorization task. To find key terms, stop words were removed and stemming was performed.

[2] http://ddm.cs.sfu.ca/dmsoft/Clustering/products/.

Table 7
Statistics for our test datasets.

Data sets	Number of documents	Number of natural clusters	Class size			The length of documents
	Total	Total	Max	Average	Min	Average
Class4	7094	4	3203	1774	1033	43
Hitech	2301	6	603	384	116	221
Re0	1504	13	608	116	11	76
Reuters	8649	65	3725	131	1	42
Wap	1560	20	341	78	5	216

Documents then were represented as TF (Term Frequency) vectors, and unimportant terms were discarded. This process implies a significant dimensionality reduction without loss of clustering performance.

The statistics of these datasets, after the document pre-processing described in Section 3.1, are summarized in Table 7. They are heterogeneous in terms of document size, cluster size, number of classes, and document distribution. The smallest document set contains 1504 documents, and the largest one contains 8649 documents. Each document is pre-classified into a single topic, i.e., a natural class. The class information is utilized in the evaluation method for measuring the accuracy of the clustering result. The detailed information (Özgür & Güngör, 2006) of these datasets can be described as follow:

- *Classic4*[3]: This document set is a combination of the four classes CACM, CISI, CRAN, and MED abstracts. Classic4 includes 3204 CACM documents, 1460 CISI documents from information retrieval papers, 1398 CRANFIELD documents from aeronautical system papers, and 1033 MEDLINE documents from medical journals.
- *Hitech*: The *Hitech* data set was derived from the San Jose Mercury newspaper articles, which are delivered as part of the Text REtrieval Conference[4] (TREC) collection. The categories of this document set are computers, electronics, health, medical, research, and technology.
- *Re0*: *Re0* is a text document dataset, derived from *Reuters*-21578[5] text categorization test collection Distribution 1.0. *Re0* includes 1504 documents belonging to 13 different classes.
- *Reuters*[5]: This document set is extracted from newspaper articles. These documents are divided into 135 topics mostly concerning business and economy. In our test, test we discarded document with multiple category labels, and the result is consisting of documents associated with a single topic of approximately 9000 documents and 50 categories. This dataset is also highly skewed.
- *Wap*: This dataset consists of 1560 Web pages from Yahoo! Subject hierarchy collected and classified into 20 different classes for the WebACE project (Han et al., 1998). Many categories of *Wap* are close to each other.

4.2. Evaluation of cluster quality: overall F-measure

The *F-measure* is often employed to evaluate the accuracy of the generated clustering results. It is a standard evaluation method for both flat and hierarchical clustering structure. More importantly, this measure balances the cluster precision and cluster recall. Hence, we define a set of document clusters generated from the clustering result, denoted C, and another set, denoted L, consisting of natural classes, such as each document is pre-classified into a single class. Both sets are derived from the same document set D. Let $|D|$ be the number of all documents in the document set D; $|c_i|$ be the number of documents in the cluster $c_i \in C$; $|l_j|$ be the number of documents in the class $l_j \in L$; $|c_i \cap l_j|$ be the number of documents both in a cluster c_i and a class l_j. Fung et al. (2002) measured the quality of a clustering result C using the weighted sum of such maximum F-measures for all natural classes according to the cluster size. This measure is called the *overall F-measure* of C, denoted $F(C)$, and is defined as follows:

$$F(C) = \sum_{l_j \in L} \frac{|l_j|}{|D|} \max_{c_i \in C}\{F\}, \qquad \text{where } F = \frac{2PR}{P+R}, \quad P = \frac{|c_i \cap l_j|}{|c_i|} \quad \text{and} \quad R = \frac{|c_i \cap l_j|}{|l_j|} \tag{4.1}$$

In general, the higher the $F(C)$ values, the better the clustering solution is.

To compute a ratio signifying how much improvement is achieved for our proposed approach, $\mathrm{F^2IHC}$, when compared to FIHC method. The *Improvement Ratio* (*IR*) is the relative value of improvements to the $F(C)$ value of $\mathrm{F^2IHC}$. In the following, we defined the *IR*:

$$IR = \frac{F(C)^{\mathrm{F^2IHC}} - F(C)^{\mathrm{FIHC}}}{F(C)^{\mathrm{FIHC}}} \tag{4.2}$$

[3] ftp://ftp.cs.cornell.edu/pub/smart/.
[4] http://trec.nist.gov/.
[5] http://www.daviddlewis.com/resources/testcollections/.

Table 8
Keyword statistics of our test datasets.

Data set	Number of terms	Number of key terms	Percentage of term removed (%)	Parameter (α threshold)	Parameter (β threshold)	Parameter (γ threshold)
Class4	41,681	41,251	1.0	0.028	0.01	0.005
Hitech	126,737	20,830	83.6	0.015	0.04	0.0008
Re0	2886	2696	6.6	0.01	0.05	0.0005
Reuters	16,641	14,679	11.8	0.05	0.06	0.003
Wap	8460	8021	5.2	0.01	0.07	0.0007

where $F(C)^{F^2IHC}$ and $F(C)^{FIHC}$ represent the $F(C)$ values of F^{2}IHC and FIHC, respectively. A higher IR value indicates that the clustering quality of F^{2}IHC method is better than the clustering quality of FIHC.

4.3. The effect of feature selection

In document clustering, feature selection is essential to make the clustering task efficient and more accurate. The most important goal of feature selection is to extract topic-related terms, which could present the content of each document.

Before applying F^{2}IHC, we first consider the feature selection strategy. To select the most representative features, we use Formulas (3.1)–(3.3) to obtain three weights and select these terms, which their weights are all higher than the pre-defended thresholds. Table 8 shows the keyword statistics of our test datasets and the suggested thresholds for each datasets.

4.4. Evaluation results

We have conducted experiments to compare the accuracy of our algorithm F^{2}IHC with other methods in Section 4.4.1. In Section 4.4.2, we further evaluate the accuracy of F^{2}IHC with respect to different MinSup parameters ranging from 2% to 9%. The efficiency of our algorithm is measured in Section 4.4.3.

4.4.1. Accuracy comparison

Table 9 presents the obtained overall *F*-measure values for F^{2}IHC and FIHC algorithms by comparing four different numbers of clusters, namely 3, 15, 30, and 60. We use the same minimum support, ranging from 3% to 6%, to test FIHC and F^{2}IHC in each data set, and list their average overall *F*-measure values.

Table 9
Comparison of the overall *F*-measure.

Data set (# of natural clusters)	# of clusters	F^{2}IHC	FIHC	UPGMA	Bi. *k*-means
Classic 4 (4)	3	0.51	0.53	N.A.	0.59*
	15	0.53*	0.53*	N.A.	0.46
	30	0.54*	0.52	N.A.	0.43
	60	0.56*	0.45	N.A.	0.27
	Average	0.54*	0.51	N.A.	0.44
Hitech (6)	3	0.47	0.48	0.33	0.54*
	15	0.47*	0.45	0.33	0.44
	30	0.48*	0.46	0.47	0.29
	60	0.45*	0.42	0.40	0.21
	Average	0.47*	0.45	0.38	0.37
Re0 (13)	3	0.55*	0.40	0.36	0.34
	15	0.54*	0.41	0.47	0.38
	30	0.54*	0.38	0.42	0.38
	60	0.54*	0.40	0.34	0.28
	Average	0.54*	0.40	0.40	0.34
Reuters (65)	3	0.49*	0.48	N.A.	0.48
	15	0.56*	0.47	N.A.	0.42
	30	0.57*	0.47	N.A.	0.35
	60	0.54*	0.42	N.A.	0.30
	Average	0.54*	0.46	N.A.	0.39
Wap (20)	3	0.39	0.37	0.39	0.40*
	15	0.61*	0.49	0.49	0.57
	30	0.62*	0.56	0.58	0.44
	60	0.62*	0.59	0.59	0.37
	Average	0.56*	0.50	0.51	0.45

N.A. means not available for large datasets.
The experimental results of UPGMA and Bi. *k*-means are the same as that of FIHC.
* The best competitor.

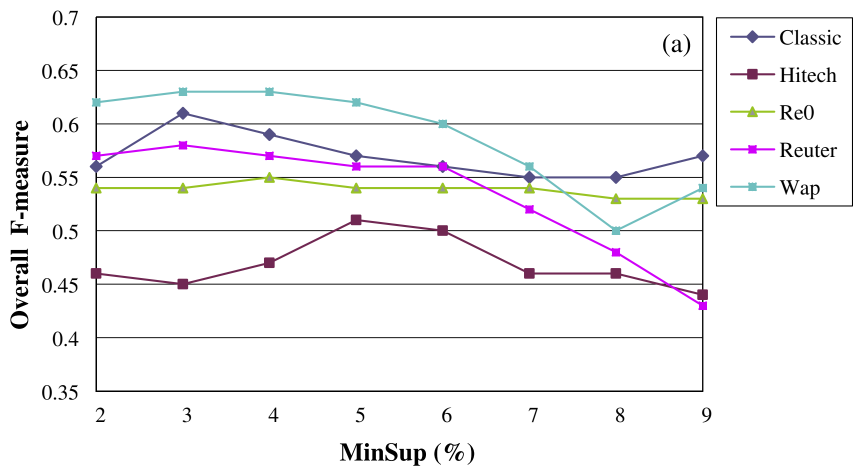
It is apparent that the average accuracy of F^2IHC is superior to that of all other algorithms. Although the performances of UPGMA, Bisecting *k*-means, and FIHC are slightly better than that of F^2IHC in several cases, we argue that the exact number of clusters in a document set is usually unknown in real case, and F^2IHC is robust enough to produce stable, consistent and high quality clusters for a wide range number of clusters. This can be realized by observing the average overall *F*-measure values of all test cases. Notice that as UPGMA is not available for large data sets because some experimental results cannot be generated for UPGMA, and we denoted them as N.A.

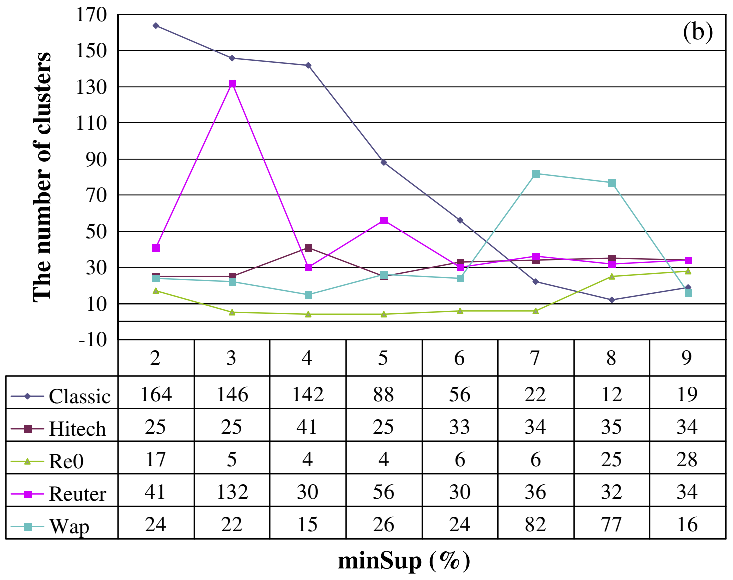
From the experimental result in Table 9, based on Formula (4.2), our proposed approach has gained (0.54–0.4)/0.4 = 35% and (0.54–0.42)/0.42 = 28% *F*(*C*) value improvement in average on Re0 and Reuters datasets, respectively, compared with FIHC algorithm. For the other datasets, the reasons for limited improvement may be due to the numbers of clusters were fixed with 3, 15, 30, and 60 for comparison purpose, and these numbers of clusters may not be appropriate for these datasets.

4.4.2. Sensitivity to various parameters

Our algorithm has two main parameters for the adjustment of accuracy quality. We now discuss how the default values were chosen, the effects of modifying those parameters, and suggestions for practical uses. The first one is mandatory and is denoted MinSup, which means the minimum support for fuzzy frequent itemset generation. The other is optional, and is denoted KCluster, which represents the number of clusters at level 1 of the cluster tree. In Table 9, we do not only demonstrate the accuracy of the produced solutions, but also show the sensitivity of the accuracy of KCluster.

Fig. 11a depicts the overall *F*-measure values of F^2IHC when accepting different mandatory parameters, but ignoring the parameter values of the optional ones. We observe that high clustering accuracies are fairly consistent while MinSup are set





	2	3	4	5	6	7	8	9
Classic	164	146	142	88	56	22	12	19
Hitech	25	25	41	25	33	34	35	34
Re0	17	5	4	4	6	6	25	28
Reuter	41	132	30	56	30	36	32	34
Wap	24	22	15	26	24	82	77	16

Fig. 11. The accuracy test of F^2IHC for different MinSup values with the optimal cluster numbers determined by the sibling merging algorithm.

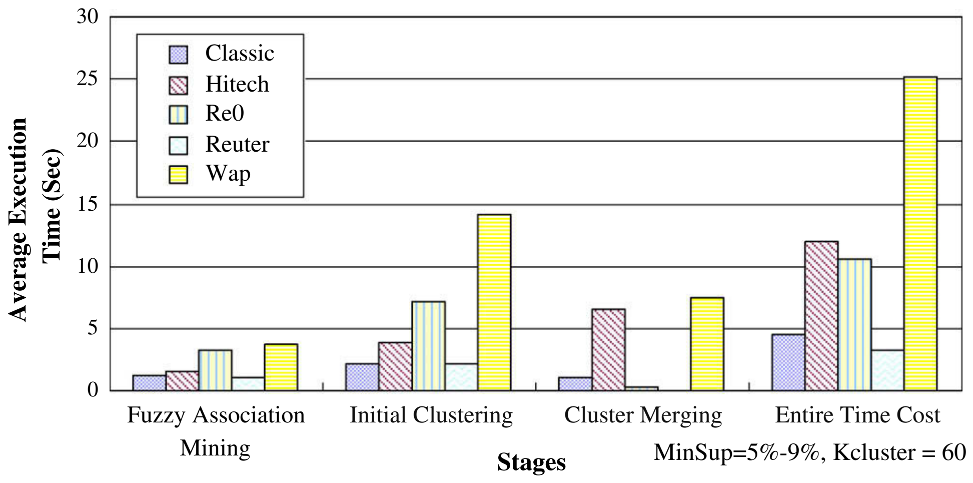


Fig. 12. The detailed time cost analysis of F^{2}IHC on five datasets.

between 2% and 9%. As KClusters is not specified in each test case, the sibling merging algorithm has to decide the most appropriate number of output clusters, which are shown in Fig. 11b.

Based on our test, we observe a general guidance that the best choice of MinSup can be set between 3% and 6%. Nevertheless, it cannot be over emphasized that MinSup should not be regarded as the only parameter for finding the optimal accuracy. It is supposed that users are responsible to adjust the shape of the cluster tree based on the value of MinSup. The smaller the value of MinSup, the deeper (and broader) a cluster tree can be generated, and vice versa.

4.4.3. Efficiency and scalability

Our algorithm involves three major phases: finding fuzzy frequent itemsets, initial clustering, and clusters merging. Fig. 12 depicts the average execution time of F^{2}IHC algorithm on five datasets, where there were five different MinSup, 5–9%, set to evaluate the performance. According to the result shown in Fig. 12, the document length dominates the performance of the execution time. From Fig. 12, we further found that the average execution time of the fuzzy mining stage on five datasets is almost identical. The runtime of our algorithm is inversely related to the input parameter MinSup. In other words, runtime increases as MinSup decreases. Due to the longer average length of documents in Hitech and Wap datasets, their average initial clustering and cluster merging time is higher than that of the other datasets.

To analyze the scalability of our algorithm, we get 100,000 documents from RVC1 (Reuters Corpus Volume 1) dataset (Lewis, Yang, Rose, & Li, 2004), which contains news from Reuters Ltd. There are three category sets: Topics, Industries, and regions. In our experiments, we consider the Topics category set, which includes 23,149 training and 781,265 testing documents. Before clustering this dataset, documents were parsed by converting all terms in documents into lower case, removing stop words, and applying the stemming algorithm.

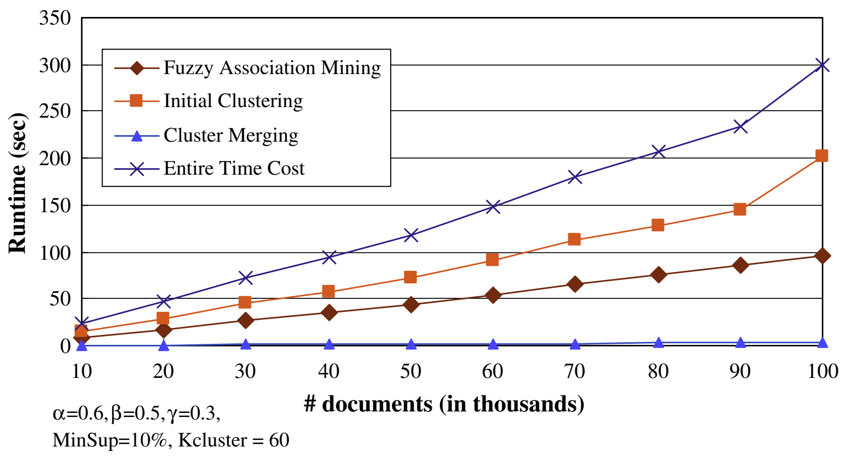


Fig. 13. Scalability of F^{2}IHC.

Fig. 13 shows the runtimes with respect to the different sizes of RVC1 dataset, ranging from 10 K to 100 K documents, for different stages of our algorithm. The whole process was completed within five minutes. The figure also shows that fuzzy mining and the initial clustering stages are the most two time-consuming stages in our algorithm. In the clustering stage, most of the time is spent on constructing initial clusters and its runtime is almost linear with respect to the number of documents.

5. Conclusions and future directions

Although numerous interesting document clustering methods have been extensively studied for many years, the high computation complexity and space need still make the clustering methods inefficient. Hence, reducing the heavy computational load and increasing the precision of the unsupervised clustering of documents are important issues. In this paper, we derived a fuzzy-based hierarchical document clustering approach, based on the fuzzy association rule mining, for alleviating these problems satisfactorily. In our approach, we start with the document pre-processing stage; then employ the fuzzy association data mining method in second stage; automatically generate a candidate cluster set, and merge the high similar clusters, and finally build a hierarchical cluster tree in a top-down fashion for easy browsing. Our experiments show that the accuracy of our algorithm is higher than that of FIHC method, UPGMA, and Bisecting *k*-means when compared on the five standard datasets. Moreover, the experiment results show that the use of fuzzy association rule mining discovery important candidate clusters for document clustering to increase the accuracy quality of document clustering. Therefore, it is worthy extending in reality for concentrating on huge text documents management.

Our future work focuses on the following two aspects:

1. *Combining the semantic analysis:* For improving the performance of the document clustering algorithm, it can be combined with semantic information, e.g., WordNet, to distinguish the senses of these key terms (Hotho, Staab, & Stumme, 2003). We will further improve F^{2}IHC algorithm with some domain knowledge, like WordNet, for higher accuracy.
2. *Incrementally updating the cluster tree:* An efficient incremental clustering algorithm for assigning new document to the most similar existing cluster should be proposed as the future direction. Since the number of documents increase sequentially in a document set, it is inefficient to reform the cluster tree with a new upcoming document. It reflects the current state of the whole document set by incrementally updating the cluster tree (Guha, Meyerson, Mishra, Motwani, & O'Callaghan, 2003; Pons-Porrata, Berlanga-Llavori, & Ruiz-Shulcloper, 2007). Moreover, some of the recent researches on data mining in steam data (Ordonez, 2003) are closely related to incremental clustering.

Acknowledgements

This research was partially supported by National Science Council, Taiwan, ROC, under Contract Nos. NSC 96-2416-H-327-008-MY2 and NSC 96-2221-E-009-168-MY2.

References

Beil, F., Ester, M., & Xu, X. (2002). Frequent term-based text clustering. In *Proceedings of the 8th ACM SIGKDD int'l conf. on knowledge discovery and data mining* (pp. 436–442).

Bellot, P., & El-Beze, M. (1999). *A clustering method for information retrieval*. Technical report IR-0199.

Chen, C. L., Tseng, Frank S. C., & Liang, T. (2008). Hierarchical document clustering using fuzzy association rule mining. In *Proceedings of the 3rd international conference of innovative computing information and control* (*ICICIC2008*) (pp. 326–330).

Delgado, M., MartÃn-Bautista, M. J., SÃanchez, D., & Vila, M. A. (2002). Mining text data: Special features and patterns. In *Proceedings of EPS exploratory workshop on pattern detection and discovery in data mining* (pp. 140–153).

Feldman, R., & Dagan, I. (1995). Knowledge discovery in textual databases (KDT). In *Proceedings of the 1st ACM SIGKDD int'l conf. on knowledge discovery and data mining* (pp. 112–117).

Fung, B. C. M., Wang, K., & Ester, M. (2002). *Hierarchical document clustering using frequent itemsets*. Master thesis, Simon Fraser University.

Fung, B. C. M., Wang, K., & Ester, M. (2003). Hierarchical document clustering using frequent itemsets. In *Proceedings of the 3th SIAM int'l conf. on data mining* (pp. 59–70).

Guha, S., Meyerson, A., Mishra, N., Motwani, R., & O'Callaghan, L. (2003). Clustering data streams: Theory and practice. *IEEE Transactions on Knowledge and Data Engineering, 15*(3), 515–528.

Han, E. H., Boley, B., Gini, M., Gross, R., Hastings, K., Karypis, G., et al. (1998). Webace: A web agent for document categorization and exploration. In *Proceedings of the 2nd int'l conf. on autonomous agents* (pp. 408–415).

Hong, T. P., Lin, K. Y., & Wang, S. L. (2003). Fuzzy data mining for interesting generalized association rules. *Fuzzy Sets and Systems, 138*(2), 255–269.

Hotho, A., Staab, S., & Stumme, G. (2003). Wordnet improves text document clustering. In *Proceedings of the semantic web workshop of the 26th annual international ACM SIGIR conference*.

Hu, G., Zhou, S., Guan, J., & Hu, X. (2008). Towards effective document clustering: A constrained *K*-means based approach. *Information Processing and Management, 44*(4), 1397–1409.

Iliopoulos, I., Enright, A. J., & Ouzounis, C. A. (2001). Textquest: Document clustering of Medline abstracts for concept discovery in molecular biology. In *Proceedings of the 6th annual Pacific symposium on biocomputing* (pp. 384–395).

Kaya, M., & Alhajj, R. (2006). Utilizing genetic algorithms to optimize membership functions for fuzzy weighted association rule mining. *Applied Intelligence, 24*(1), 7–15.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research, 5*, 361–397.

Lin, S. H., Shih, C. S., Chen, M. C., Ho, J. M., Ko, M. T., & Huang, Y. M. (1998). Extracting classification knowledge of internet documents with mining term association: A semantic approach. In *Proceedings of the 21st annual int'l ACM SIGIR conf. on research and development in information retrieval* (pp. 241–249).

Mandhani, B., Joshi, S., Kummamuru, K. (2003). A matrix density based algorithm to hierarchically co-cluster documents and words. In *Proceedings of the 12th int'l conf. on World Wide Web* (pp. 511–518).

Martín-Bautista, M. J., Sánchez, D., Chamorro-Martínez, J., Serrano, J. M., & Vila, M. A. (2004). Mining web documents to find additional query terms using fuzzy association rules. *Fuzzy Sets and Systems, 148*(1), 85–104.

Ordonez, C. (2003). Clustering binary data streams with *k*-means. In *Proceedings of the 8th ACM SIGMOD workshop on research issues in data mining and knowledge discovery* (pp. 2–19).

Özgür, A., & Güngör, T. (2006). Classification of skewed and homogeneous document corpora with class-based and corpus-based keywords. In *Proceedings of the 29th German conf. on artificial intelligence* (pp. 91–101).

Pons-Porrata, A., Berlanga-Llavori, R., & Ruiz-Shulcloper, J. (2007). Topic discovery based on text mining techniques. *Information Processing and Management, 43*(3), 752–768.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Salton, G. (1971). *The SMART retrieval system – Experiments in automatic document retrieval, Retrieval*. Englewood Cliffs, NJ: Prentice Hall.

Shahnaz, F., Berry, M. W., Pauca, V. P., & Plemmons, R. J. (2006). Document clustering using nonnegative matrix factorization. *Information Processing and Management, 42*(2), 373–386.

Shihab, K. (2004). Improving clustering performance by using feature selection and extraction techniques. *Journal of Intelligent Systems, 13*(3), 135–161.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. In *Proceedings of the KDD workshop on text mining*.

Tan, A. H. (1999). Text mining: The state of the art and the challenges. In *Proceedings of the Pacific Asia conf. on knowledge discovery and data mining* (pp. 65–70).

Xu, W., & Gong, Y. (2004). Document clustering by concept factorization. In *Proceedings of the 27th ACM SIGIR conf. on research and development in information retrieval* (pp. 202–209).

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*, 338–353.

Chun-Ling Chen is a Ph.D. student in Dept. Computer Science, National Chiao-Tung University, Taiwan, ROC. Her research interests include database, object-oriented conceptual modeling, information retrieval, text mining, and fuzzy set theory.

Frank S.C. Tseng received his B.S., M.S. and Ph.D. Degrees, all in computer science and information engineering, from National Chiao Tung University, Taiwan, ROC, in 1986, 1988, and 1992, respectively. He joined the faculty of the Department of Information Management, Yuan-Ze University, Taiwan, ROC, on August 1995. From 1996 to 1997, he was the Chairman of the Department. Currently, he is a professor and the chairman of the Department of Information Management, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan, ROC. His research interests include database theory and applications, information retrieval, XML technologies for Internet computing, data/document warehousing, and data/text mining. Dr. Tseng is a member of the IEEE Computer Society and the Association for Computing Machinery.

Tyne Liang received her Ph.D. Degree from National Chiao Tung University, Taiwan, ROC, majored in computer science. Currently, she is an associate professor of the Dept. of Computer Science, National Chiao Tung University, Taiwan, ROC. Her research interests include information retrieval, natural language processing, web mining, and inter-connection network.