A Fuzzy based Document Clustering Algorithm

Kabita Thaoroijam
IIIT Manipur
Imphal, Manipur
India

A. Kakoti Mahanta
Gauhati University
Guwahati, Assam
India

ABSTRACT
Document clustering is an automatic grouping of text documents into clusters so that documents within a cluster have high similarity values among one another, but dissimilar to documents in other clusters. It has wide applications in areas such as search engines, web mining, information retrieval and topological analysis. This paper presents a new document clustering algorithm using the concept of fuzzy sets, where each cluster is viewed as a fuzzy set over some finite universal set. The algorithm was implemented and the results are reported. The efficiency and time complexity of the algorithm have also been discussed.

General Terms
Data Mining, Clustering

Keywords
Document Clustering, Fuzzy Set, Agglomerative Algorithm, Compact Representation.

1. INTRODUCTION
Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters) [1]. The goal of clustering is to classify data from a given dataset into groups such that the datapoints within a group are more similar to each other than to those outside the group. Text databases are rapidly growing due to the increasing amount of information available in electronic forms, such as electronic publications, e-mails and the WWW. Most researchers of datamining have focused on structured data. Data stored in most text databases are semi-structured or unstructured, thus the modeling and implementation of semi-structured or unstructured data have become an essential part of document mining. Given a collection of unlabelled documents, document clustering can help in organizing the collection, thereby facilitating future navigation and search. Document clustering is useful in many information retrieval tasks such as document browsing, organization and viewing of retrieval results, generation of hierarchies of documents in search engines etc. In [8] a comparative study of common document clustering techniques is done.

Unlike document classification, no label documents are provided in clustering; hence clustering is also known as unsupervised learning. Most document clustering methods perform several preprocessing steps including stopword removal and stemming on the document set. Each document is represented by a vector of frequencies of the remaining terms within that document. There are many basic reasons for interest in unsupervised learning. Collecting and labeling a large dataset of sample patterns can be surprisingly costly. In early stages of an investigation it may be valuable to perform exploratory data analysis and thereby gain some insight into the nature or structure of the data. By clustering one can identify dense and sparse regions and therefore, discover distribution patterns and interesting correlations among data attributes. Unfortunately, although there are various traditional clustering techniques, but they cannot be applied for clustering text data due to the basic properties of text databases:

- volume of the input database
- high dimensionality of feature set
- sparseness in document vector
- complex and ambiguous semantics and
- noisy data.

In this paper, a new document clustering algorithm using the concept of fuzzy sets is proposed. In the proposed algorithm at any given stage of the algorithm there are small clusters and the decision at the current stage is to merge the incoming document with the cluster that satisfies a user defined threshold. The algorithm is agglomerative. The clusters obtained are represented as fuzzy sets over a finite universal set. A similarity measure based on the fuzzy representation of the clusters is defined. The algorithm requires just one pass through the dataset and only the compact representations of the clusters are kept in the memory at any given time. The algorithm starts by considering each input data point as a cluster, compares it with the existing clusters at that stage of the algorithm and is merged with the cluster that satisfies a user defined threshold. In section 2, some recent and similar works on clustering using fuzzy approach is discussed. Section 3 describes the method of representing a cluster as a fuzzy set, the similarity measure used and the merge function used to merge pairs of clusters. In section 4, the proposed algorithm and its complexity is discussed. In section 5, the experimental result is reported.

2. RELATED WORK
During the last few years the concept of fuzzy sets has been used in different areas including clustering or pattern recognition ([3], [9], [10], [11]). Conventional clustering techniques assume that an object or data point can belong to one and only one cluster. However there may be overlapping of clusters and thus the separation of clusters is a fuzzy notion and hence the concept of fuzzy sets has come into picture. In fuzzy clustering each data point is associated with each cluster using a membership value. Larger membership values indicate higher confidence in the assignment of the object to the cluster. So in this approach each cluster is a fuzzy set of all data points.

In the paper [6] the authors proposed an approach of fuzzy clustering of web documents. The documents are represented as vectors of variable lengths. Each element of the vector is a pair of key phrase and an importance weight associated with this key phrase in a particular document. Using this representation of documents, fuzzy clustering algorithm was applied.

In [5] the authors proposed a fuzzy set approach for clustering large categorical data. For study of clusters, the underlying dataset was considered as a market-basket dataset where each transaction is a set of items bought by a particular customer. If

I be the set of all items under consideration, each point in a cluster is a subset of I and each cluster is a collection of such subsets. In this context the clusters were defined as fuzzy sets over the set of all items.

3. CLUSTERS AS FUZZY SETS

3.1 Fuzzy Set Representation of Clusters

After some preprocessing step is applied to the documents, each document is represented as a finite list where each element in the list is of the form (w : number of occurrence of w in the document) for each distinct keyword w occurring in the document. Let W be the set of all distinct key-words appearing in the documents under consideration. Let $|W| = m$. The keywords are numbered in some order and thus get a sequence of the form $W = \{w_1, w_2, w_3, ...w_m\}$. Now keeping this ordering in mind, any document d can be represented as the m-tuple $(o_1, o_2, o_3, ...o_m)$ where o_i indicates the number of occurrence of the word w_i in the document d. If a word w_i is not present in a document then $o_i = 0$ for that document. In this context each cluster is defined as a fuzzy set over W. The fuzzy set representation of the cluster C consisting of only one document say d represented as the m-tuple $(o_1, o_2, o_3, ...o_m)$ is computed as follows. Let F_C be the fuzzy set and μ_{F_C} be the associated membership function, then $\mu_{F_C} : W \to [0,1]$ is defined as

$$\mu_{F_C}(w_i) = \frac{o_i}{\sum_{i=1}^{m} o_i} \tag{1}$$

Obviously $0 \le \mu_{F_C}(w_i) \le 1$ for each i. The fuzzy set F_C represented by the membership function μ_{F_C} together with $o_{sum} = \sum_{i=1}^{m} o_i$ is a compact representation of the cluster C and this is how the clusters are represented. In Section 3.3 it is shown how to obtain this representation for the cluster obtained by merging two clusters whose compact representations are given. So the compact representation of a cluster C is represented as (F_C, o_{sum}).

3.2 Merging of Clusters

Let C_1 and C_2 be the two clusters and o_{sum1} and o_{sum2} be the total number of terms appearing in C_1 and C_2 respectively. Let $(\mu_{F_{C_1}}, o_{sum1})$ and $(\mu_{F_{C2}}, o_{sum2})$ be their compact representations. Let C be the cluster obtained by merging C_1 and C_2, and let F_C be the fuzzy set representing C. Then the Fuzzy membership function μ_{F_C} for F_C can be computed as follows

$$\mu_{F_C}(w_k) = \frac{\left(o_{sum1}\mu_{F_{C_1}}(w_k) + o_{sum2}\mu_{F_{C2}}(w_k)\right)}{(o_{sum1} + o_{sum2})} \tag{2}$$

for $k = 1, 2, ..., m$. Thus the compact representation of C is (F_C, o_{sum}) where $o_{sum} = o_{sum1} + o_{sum2}$.

3.3 Similarity Measure Between 2 Clusters

A similarity function of pairs of clusters is defined which can be calculated from the fuzzy set representation of the clusters. Let C_1 and C_2 be two clusters and let F_{C_1} and F_{C_2} be the fuzzy sets representing C_1 and C_2 respectively. Let $sim(C_1, C_2)_{fuzzy}$ be the value of the similarity function, then

$$sim(C_1, C_2)_{fuzzy} = \frac{|F_{C_1} \cap F_{C_2}|}{|F_{C_1} \cup F_{C_2}|} \tag{3}$$

where union, intersection and cardinality of fuzzy sets are computed as defined in [2].

The cosine measure is also used as a similarity function. It is computed as

$$sim(C_1, C_2)_{\cos ine} = \frac{(F_{C_1} \cdot F_{C_2})}{\|F_{C_1}\| \|F_{C_2}\|} \tag{4}$$

4. PROPOSED ALGORITHM

4.1 Preprocessing Step

All document clustering methods require several steps of preprocessing of the input data before performing the actual clustering. Firstly the non-textual elements from the documents are removed. Stopword removal was done using a standard stopword list. Then a master word list containing every word in the document set, associated with its overall frequency is created. The master word list is cut down by removing infrequent words. In each document, words that do not appear in the master word list are removed. Finally, a feature vector was created for each document where each element had two fields. The first being the word present in that document and the second is the frequency of the word in the document. After the preprocessing step each document is represented as a finite list where each element in the list is of the form (w : number of occurrence of w in the document) for each distinct keyword w occurring in the document. The ordering of the words in the document is maintained.

4.2 The Proposed Algorithm

The algorithm accepts as input the following:

- The value of n, which is the size of the input data set.
- The n input data points (i.e. documents after the preprocessing step)
- The value of θ which is the threshold used for merging clusters

Let S denote the set of clusters obtained at any time during the execution of the algorithm. Each cluster in S is represented as fuzzy set as described in the previous section. The algorithm is described below.

```
begin
  set S = φ
  input n , θ
  for i = 1 to n do
  begin
    input a data point d
    compute the cluster C consisting
            of the data point d only
```

```
    while there is C1 ∈ S with sim(C,C1) ≥ θ
    begin
        C2 ← merge(C,C1)
        remove C1 from S
        delete cluster C
        C ← C2
    end
    add C to S
  end
end
```

The set S gives the final set of clusters.

In the algorithm the function $sim(C,C_1) \geq \theta$ is as described in Equations (3) and (4). The function $merge(C,C_1)$ gives a new cluster after merging the clusters C and C_1 as described in equation (2). The algorithm requires just one pass through the database and it is not necessary to keep the data points in memory. Only the summary of the clusters using fuzzy sets are kept in memory at any time.

4.3 Complexity of the Algorithm

Let n be the total size of the input data set. The complexity of computing the summary of a cluster containing 1 element is $O(m)$ where m is the size of the feature vector to which the input data points are converted. In the whole algorithm n such computations are necessary. Thus the overall complexity of this process is $O(mn)$. The complexity of computing the similarity value between a pair of clusters is $O(m)$.In the algorithm at most n^2 such computations will be needed. Thus the complexity of this process is $O(mn^2)$. The complexity of the procedure of merging two clusters using cluster summary is $O(m)$. During the execution of the whole algorithm at most $n-1$ times this procedure will be executed. Thus the overall complexity of the algorithm is $O(mn+mn^2)$ i.e. $O(mn^2)$.

5. EXPERIMENTAL RESULTS

Three datasets were used for the experimental evaluation. The datasets are BankSearch, 20NewsGroup and Reuters21578.

- The BankSearch dataset [7] is a collection of 11,000 documents divided into 11 categories with each category having 1000 documents. 3 subsets of the BankSearch dataset were used, namely: (i) ADJ consisting of 3 distinct categories namely Banking and finance, Programming language and Sports, (ii) ABC consisting of 3 categories on related theme of Banking and Finance and (iii) ADGHJ consisting of 5 distinct categories of Banking and finance, Programming language, Astrology, Biology and Sports.
- The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. Each of the 20 newsgroup topics contains roughly 1000 postings and it was originally collected by Ken Lang. Two subsets of the dataset were used: (i) A2 consisting of alt.atheism and comp.graphics and (ii) B2 consisting of talk.politics.guns and talk.politics.mideast
- The Reuters-21578 dataset is a collection of 21578 documents that appeared on Reuters news service in the year 1987. Only one subset of the Reuters-21578 dataset have been taken consisting of the topics (i) coffee, (ii) gold, (ii) interest, (iv) ship and (v) sugar. The primary topic keyword is used as the category.

The proposed algorithm was implemented and tested on the datasets mentioned above. The clustering result of the proposed algorithm is compared with the result of k-means algorithm. The clustering results are evaluated using rand index [4] which is shown in table 1. The proposed algorithm performed better than k-means algorithm on the given datasets. The proposed similarity measure performed better or almost same compared to cosine measure both in the case of k-means algorithm and the proposed algorithm.

It is noticed that by increasing the similarity threshold value, the quality of the clusters obtained improves to a high extent. Majority of the clusters obtained were pure clusters. But the clusters obtained were of small sizes, thereby producing a large number of clusters.

Table 1. Performance of proposed Algorithm and K-means

Dataset	k-means		Proposed Algorithm	
	Cosine Measure	***Fuzzy Similarity Measure***	***Cosine Measure***	***Fuzzy Similarity Measure***
ADJ	0.76296	0.75494	0.86221	0.80213
ABC	0.62575	0.62723	0.68574	0.73207
ADGHJ	0.70324	0.70227	0.81798	0.80593
A2	0.61972	0.63288	0.75582	0.76484
B2	0.61576	0.65297	0.71836	0.77489
Reuters	0.70262	0.71534	0.72358	0.80338

6. CONCLUSION

In this paper, a fuzzy based algorithm for clustering document collections is presented. With the dynamic nature of real world data, any algorithm must be able to deal with new data that is constantly added to the databases. Since the proposed algorithm is incremental, whole database need not be used for clustering every time when the database is being updated. The results of the experimental study are quite encouraging with the algorithm classifying with good accuracy. In the future attempts will be made to work out a method for merging the relatively large number of small clusters to form bigger ones.

7. REFERENCES

[1] A. K. Jain, M. N. Murty, and P.J. Flynn, "Dataclustering: A review", ACM Computing Surveys, 31(3): 264-323, 1999.

[2] I. Ludmila Kuncheva, Fuzzy Classifier Design, Physica-Verlag.

[3] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithm", Plenum Press, New York

[4] K. Yeung and W. Ruzzo, Details of the adjusted rand index and clustering algorithms, supplement to the paper "an experimental study on principal component analysis for clustering gene expression data". Bioinformatics (17), 763-774, 2001.

[5] M. Dutta and A. Kakoti Mahanta, "An algorithm for clustering large categorical databases using a Fuzzy set based approach", Proceedings of NWTAC (National Workshop on Trends in Advanced Computing) 2006, Tezpur University.

[6] M. Friedman, M. Last, O. Zaafrany, M. Schneider, and A. Kandel, "A New Approach for Fuzzy Clustering of Web Documents", Fuzzy Systems, Proceedings. 2004 IEEE International Conference, Vol 1, 377- 381, July 2004

[7] M. P. Sioka and D. W. Come, "The BankSearch web document dataset: investigating unsupervised clustering and category similarity", Journal of Network and Computer Applications Volume 28, Issue 2 (April 2005).

[8] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques", Proc. KDD-2000 Workshop on TextMining, Aug. 2000.

[9] R. N. Dave, "Generalized fuzzy C-shells clustering and detection of circular and elliptic boundaries", Pattern Recognition, 25,713-722

[10] S. K. Pal, "Fuzzy tools for the management of uncertainty in pattern recognition, image analysis, vision and expert systems", International J. System Sc, Vol 22,No 3, pp 511-549, 1991.

[11] W. Pedrycz, "Fuzzy sets in pattern recognition: Methodology and methods", Pattern Recognition, Vol 23, No ½, pp121-146, 1990