Incremental Document Clustering Using Multi-representation Indexing Tree

Lifeng Wang, Hui Song, Xiaoqiang Liu
Department of computer science and technology
Donghua University
Shanghai, China
lfwang@mail.dhu.edu.cn, songhui@dhu.edu.cn, liuxq@dhu.edu.cn

***Abstract*—Incremental Document Clustering is a powerful technique for large-scale topic discovery from incremental documentation set. Indexing tree algorithm is advanced in efficiency. However, it tended to process spherical data. To address this problem, we present a novel Multi-Representation Indexing Tree (MRIT) algorithm for constructing a hierarchy that satisfies arbitrary shape clusters with a good performance. Compared with the Indexing tree algorithm, a cluster is decomposed into several sub clusters and is represented as a union of the sub clusters rather than the center of the cluster. Similarity of a document to one cluster is the distance to the nearest neighbor among the cluster's representative points. The experimental results on a variety of domains demonstrate that our algorithm can produce a quality cluster. It's insensitive to document input order, and efficient in terms of computational time.**

Keywords-Incremental Clustering; MRIT; Indexing Tree; Multi-representation

I. INTRODUCTION

Clustering of large collections of text documents is a key process in providing a higher level of knowledge about the underlying inherent classification of the documents. With the advent of online publishing in the World Wide Web, the number of documents generated everyday has increased considerably, incremental clustering emerged.

Compared with batch clustering, incremental method only deals with the new documents added to the data set, avoiding re-clustering the entire data from scratch. In spite of the potential benefits of the incremental algorithm, there lacks accurate and efficient in practical text document clustering application [1], a lot of researches focus on these issues recently [2,3,4].

In this paper, we present an incremental clustering algorithm based on dynamic Indexing tree [2], represented as a Multi-Representation Indexing Tree (MRIT), which is a concise statistical Multi-representation of the Indexing tree. And define similarity metric as the shortest distance between a document and the nearest neighbor (among the cluster's representing points), which aims to prevent decentralizing of the cluster. Experimental results show our idea achieve better effectiveness and efficiency.

The rest of this paper is organized as follows. We discuss related work on incremental clustering in section 2. In section 3, we briefly introduce the fundamental of dynamic Indexing tree. We explain the contributions of our work, the MRIT algorithm in section 4. Section 5 gives our experimental results, and conclusions with future work are wrapped up in Section 6.

II. RELATED WORK

There are many published algorithms, which aimed to incrementally cluster points in a data set, including DC-tree clustering[5], incremental hierarchical clustering [6], et al.

Widyantoro's [3] approach is more related to the agglomerative hierarchical clustering techniques [7, 8], and therefore, It can be viewed as the incremental version of the traditional bottom-up hierarchical clustering methods.

Khaled M. Hammouda [4] proposed SHC incremental clustering, which relies only on pair-wise document similarity information. Clusters are represented with a Cluster Similarity Histogram. A concise statistical representation of the distribution of similarities within each cluster provides a measure of cohesiveness. However, the time complexity of SHC is $O(n^2)$, since it must compute the similarity to all previously seen documents for each new one.

Chung-Chian Hsu [9] proposed M-ART and the conceptual hierarchy tree to solve similar degrees of mixed data.

Zhang Kuo [2] represented an incremental clustering based on the news indexing-tree created dynamically. Indexing-tree is created by assembling related documents together to form news clusters in different hierarchies according to their values of similarity. Comparisons between current document and previous clusters could help finding the most similar document in less comparing times. It performs the clustering process efficiently with $O(n)$ time complexity. However, it prefers spherical data and leads to lower accuracy caused by decentralization of class center. Our work contributes on improving the accuracy without increasing computation time.

III. DAYNAMIC INDEXING-TREE

A. Pre-Processing and Page Representation

The primary step of text clustering is the extraction of features from documents to create a term vector for each document, followed by clustering or grouping based on those features.

Incremental TF-IDF model is widely applied in term weight calculation. TF-IWF model [10] is chosen to weight

This work was supported by the National Natural Science Foundation of China (No. 60903160).

978-1-4244- 7618-3 /10/$26.00 ©2010 IEEE

terms for its steadier performance in many experiments. WF (word frequency) of the term w at time t is calculated as:

$$wf_t(w) = wf_{t-1}(w) + wf_{St}(w) \quad (1)$$

where S_t means a set of documents coming at time t, and $wf_{St}(w)$ means the appearance number of term w appears in the newly appearing documents. $Wf_{t-1}(w)$ represents the appearance number of term w appears before time t. As showed in formula (1), WF is updated dynamically at time t.

Each document d coming at time t is represented as an n-dimension vector, where n is the number of distinct terms in document d. Each dimension is weighted using incremental TF-IWF model, and the vector is normalized so that it is of unit length:

$$weight_t(d,w) = \frac{tf(d,w)\log(W_t+1)/(wf_t(w)+0.5))}{\sqrt{\sum_{w'\in d}(tf(d,w')\log((W_t+1)/(wf_t(w')+0.5)))^2}} \quad (2)$$

where $tf(d,w)$ means how many times the term w appears in document d and W_t represents the total appearance number of the term before time t:

$$W_t = \sum_{t_d \le t}\sum_{w'\in d} tf(d,w') \quad (3)$$

t_d in formula (3) means that the document d appears at time t.

B. Similarity Calculation

The cosine between the document vector and the cluster's centroid vector is used to compare a document to a cluster. To prevent longer documents from dominating centroid calculations, normalizing all document vectors to unit length is needed.

As to document d and d' at time t, their similarity is calculated as:

$$similarity_t(d,d') = \sum_{w\in d\cap d'} weight_t(d,w) * weight_t(d',w) \quad (4)$$

C. Incremental Clustering using Indexing Trees

Similar documents are put together to form a hierarchy of clusters. Similar documents are indexed together by their common ancestor. Dissimilar documents are indexed in different clusters. A cluster is a non-terminal node, and a document is a terminal node. The cluster is represented as the centroid of its descendants, and its out degree is at least 2.

Incorporating a new data point into an Indexing tree incrementally can be divided into two stages. During the first stage, it uses comparisons between the current document and previous hierarchical clusters to help finding the most similar document which is useful for new cluster decision. At the second stage, the current document is inserted to the indexing-tree. A document will be a new cluster if the similarity between it and a cluster exceeds threshold ($\theta_{init}+(h-1)\delta$, h is the length between the cluster or document and the root of the tree) of the cluster; otherwise, it will be a descendant of the cluster.

Clusters located on same levels of the Indexing tree share common threshold, so this algorithm prefers spherical data. Let us assume that there is a document A far from the center of cluster C, D_{AC} is represented as the distance from the center of C to A, and D_{AC} is equal to threshold of C. When a new document B is coming, and B is very similar to A, but $D_{BC} > D_{AC}$, it judges A and B are not in a same class. On the other hand, the order of input documents is arbitrary, a cluster will be decentralized after a set of documents with highest similar inserted into it, and few existing documents are far from them.

IV. MULTI-REPRESENT POINTS INDEXING TREE

To deal with non-spherical data, we proposed MRIT, which is a tree like a diagram to reveal the clustering hierarchy. Each non-terminal node represents a union of similar documents, and terminal node represents a single document. For avoiding the center of the cluster is decentralized, we choose several sub clusters of the cluster as its representative points.

A. Model representation

Multi-Representation Indexing Tree is defined formally as follows:

$$MRIT = \{r, N_{nc}, N_d\}$$

where r is the root of MRIT, N_{nc} is the set of all cluster nodes, N_d is the set of all document nodes. We define a set of constraints for a MRIT:

- $\forall N_i \in N_{nc} \rightarrow N_i$ is a non-terminal node in the tree
- $\forall N_i \in N_d \rightarrow N_i$ is a terminal node in the tree

A MRIT is shown as follows in Fig. 1, N_{23} is a terminal node, and N_2 is a non-terminal node.

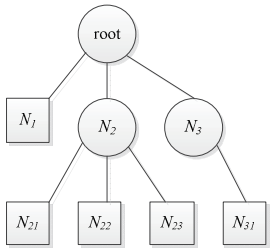

Figure 1. A sample of MRIT

and we define *Node* as follows:

$$Node=\{N_c,N_p,N_r\}$$

where N_c is a set of the node's children, N_p is its parent node, N_r is a set of representative points of the node. If N_c is empty, N_r is itself in N_l, and N_r is a subset of its direct children in N_{nl}.

B. Representative Points Selection

The representative point of the terminal node is itself.

For non-terminal node, with a subset of objects (X_i) forming, the node can be represented (or typified) by the representative points (N_r), denoted by

$$X_i \Leftrightarrow N_r \tag{4}$$

From Fig.2, we can know that nodes N_{ri} (i=1, 2, 3) are representative points of the Cluster, C is the center of the Cluster.

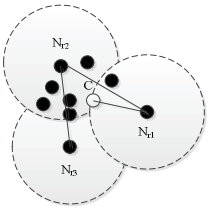

Figure 2. Multi-representation of non-terminal node

Selection can be achieved by picking up the farthest document to the previous representative points or the center of clusters to represent individual clusters the selection process can be formalized as:

$$\max_i \{| x_i - N_r \cup N_c |\} \tag{5}$$

Fig. 3 gives the pseudocode of representative points selection.

```
Function SelectMultiReprestation(C, REPREMAX)
 // C: a cluster
 // REPREMAX: the maximum number of representative points
  M = the center of C;
  For i=0 to REPREMAX
    FOREACH node IN c
      IF i==0 THEN
        get representative point which is farthest from center C
      ELSE
        get representative point which is farthest from representative points
 of C
      END IF
    NEXT
    Represents+=represent
  NEXT
  Return Represents
 END FUNCTION
```

Figure 3. Multi-representation selection algorithm

C. Similarity Calculation

We get the similarity between documents and clusters by calculating the shortest distance between documents and representative points of clusters using a cosine measurement (6).

$$similarity(d, N) = \min_{N_r \in N}(similarity(d, N_r)) \tag{6}$$

To determine whether a document can be attributed to an old cluster or be a new cluster, we give several definitions:

Definition 1 Let C be a terminal node. Given a new point A, let d be the distance from A to C. A is said to form a higher dense region in C if $d>\theta$. (θ is user-defined)

Definition 2 Let N_r be a multi-representation of a non-terminal node C. Given an upper limit $U_L = \max_{N_r \in N}\{similarity(N_r, N_c)\}$ (N_c is the center of the cluster C), the cluster C is homogeneous only if $d_i \leq U_L$ for $\forall d_i \in C$.

Definition 3 Let C be a non-terminal node. Given a new point A, let B be a representative point of C, which is the nearest neighbor to A. Let d be the distance from A to B. A is said to form a higher dense region in C if $d>L_L$.

In Fig. 4, because $d_{EC}<L_L$, so E will be added to cluster and D will be out of the cluster because $d_{DC}>L_L$.

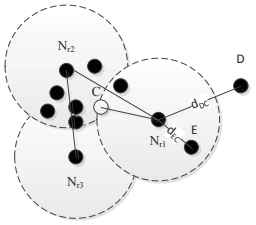

Figure 4. relationship between a new document and an cluster

D. Clustering Process

For a new document d, the comparison procedure and inserting procedure based on MRIT are defined as follows. An example is shown Fig. 5.

Comparison procedure:

Step 1: compare d to all the direct child nodes of *root* and select λ nodes with highest similarities, e.g., N_2 and N_3 in Fig. 4.

Step 2: for each selected node in the step 1, e.g. N_2, compare d to all its direct child nodes, and select λ nodes with highest similarities, e.g. N_{23} and N_{22}. Repeat step 2 for all non-terminal nodes.

Step 3: record the terminal node with the highest similarity to d, e.g. N_{23}, and the similarity value (0.9).

Insertion procedure:

Insert d to the MRIT with r as following:

Find the node n which is a direct child of the root in the path from the root to the terminal node N_{23} which has the highest similarity, e.g. N_2. If the similarity s between N_2 and d is smaller than L_L , then add d to the tree as a child of the root. Otherwise, if n is a terminal node, create a cluster node instead of n, and add both n and d as its direct children; if n is a non-terminal node, repeat this procedure and insert d to the sub-tree with n as root recursively.

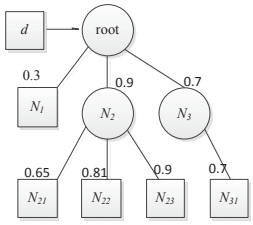

Figure 5. Comparison and insertion

Comparing with Indexing tree, we select representative points (maximum k) to describe a cluster rather than its center, so consumes k times in computing a new document's belonging. For Indexing tree's time complexity is $O(n)$, MRIT's is $O(n*k)$. In practical, k << n, our approach got the same performance in efficiency as the former.

V. EXPERIMENT

A. Datasets and Experimental Setup

In order to verify the quality of the proposed algorithm, we use the SOGOU documents set and select 2,000 documents, including many aspects, e.g. educations, sports, etc. as our test data. System-1 is implemented based on the origin indexing tree clustering; System-2 is implemented base on the approach present in section 4.

B. Parameter Selection

CF- Feature [11] is a clustering quality evaluation method. Large CF-Feature is, the better result is. We got largest CF-Feature when we set $\theta=0.5$ in the train set, so we set $\theta=0.5$ in following experiments.

C. Experiment Result

We evaluated both systems' clustering quality with three metrics: accuracy, CF-Feature and processing time.

The clustering accuracy is used as a measure of the result. It is defined as $\varphi = \sum_{i=1}^{k} x_i / N$. x_i is the number of object occurring in both the i_{th} cluster and its corresponding class, N is the number of objects in the dataset. k is the resultant number of clustering. Table I shows the accuracy of system-1 and system-2.

TABLE I. ACCURACY ON SYSTEM-1 AND SYSTEM-2

Number of documents	accuracy	
	System-1	***System-2***
100	75%	80%
500	83.42%	87.16%
2000	89.46%	93.46%

Table II shows the CF-Feature of System-2 is larger than that of System-1, when the number of documents exceeds 1000, so for large data set, System-2 is better.

TABLE II. CF-FEATURE BETWEEN CLUSTERS ON SYSTEM-1 AND SYSTEM-2

Number of documents	CF-Feature	
	System-1	***System-2***
100	0.50624	0.48285
500	0.40597	0.40129
1000	0.29695	0.29721
1500	0.27473	0.27669
2000	0.26211	0.26671

To process 2000 documents, system-1 and system-2 consume 455,522 and 219,995 milliseconds respectively, while we set REPREMAX as 5. It satisfies the theoretical analysis.

VI. CONCLUSIONS

For incremental text clustering, the Indexing tree algorithm is more efficient, but it prefers spherical data. The MRIT clustering algorithm is presented, which takes several sub-cluster to represent a cluster and use the max distance between representative points and the center of the cluster as the threshold of a cluster. The clustering results will no longer prefer a spherical shape. Theoretical analysis and experimental results show that the algorithm cannot only avoid decentralizing the center of a cluster, but also describe the characteristics of the data set accurately. Next step, we will take some measures in several respects for accurate improvement. E.g. re-structuring nodes in order to merge or divided previous cluster.

REFERENCES

[1] Nachiketa Sahoo, Jamie Callan, Ramayya Krishnan, George Duncan , Rema Padman, Incremental hierarchical clustering of text documents, Proceedings of the 15th ACM international conference on Information and knowledge management, USA 2006.

[2] K. Zhang, J. Li, and G. Wu. New Event Detection Based on Indexing-tree and Named Entity. In Proceedings of the 30th Annual International ACM SIGIR Conference, Amsterdam, the Netherlands. ACM Press. 2007, 215-222.

[3] Widyantoro, D.H., Loerger, T.R., Yen, J. An Incremental Approach to Building a Cluster Hierarchy. ICDM'02, 705-708,2002.

[4] K. Hammouda and M. Kamel. Incremental document clustering using cluster similarity histograms. IEEE/WIC International Conference on Volume, 2003.

[5] W. Wong and A. Fu. Incremental document clustering for web page classification. In 2000 Int. Conf. on Information Society (IS2000), Japan, 2000.

[6] M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental clustering and dynamic information retrieval. In *The 29th annual ACM symposium on Theory of computing*, pages 626-635, 1997.

[7] B. S. Everitt, S. Landau, and M. Leese. *Cluster Analysis.* New York, NY: Oxford University Press Inc, 2001.

[8] Wim De Smet, Marie-Francine Moens. An Aspect Based Document Representation for Event Clustering. *Proceedings of the 19th Meeting of Computational Linguistics in the Netherlands*, 2009, 55-68

[9] Chung-Chian Hsu, Yan-Ping Huang. Incremental clustering of mixed data based on distance hierarchy. in Expert Systems with Applications, vol. 35. http://www.sciencedirect.com: Elsevier Science Ltd., 2008, 1177 – 1185.

[10] C. Wang, M. Zhang, S. Ma and L. Ru. Automatic Online News Issue Construction in Web Environment. In processdings of the 17th international conference on World Wide Web, 2008, 457-466.

[11] Han xi-wu, Zhao Tie-jun. "An evaluation method for clustering quality and its application," Journal of harbin institute of technology, vol 41, pp. 225-227, November 2009, 225-227.