Middle-East Journal of Scientific Research 24 (Techniques and Algorithms in Emerging Technologies): 496-501, 2016
ISSN 1990-9233;

DOI: 10.5829/idosi.mejsr.2016.24.TAET23630

Review on Cover Coefficient Concept and Incremental Approach for Document Clustering

[1]N. Kannammal, [2]Dr. S. Vijayan and [3]R. Sathish Kumar

[1]Assistant Professor, Surya Engineering College, Erode, Tamilnadu, India
[2]Professor, Surya Engineering College, Erode, Tamilnadu, India
[3]Assistant Professor, Hindusthan College of Engineering and Technology, Coimbatore, Tamilnadu, India

Abstract: The dynamic growth of web service influence the quality of web discovery and selection for a request. Clustering is a data mining technique that group services based on several characteristics (functional, non-functional, user rating, item ratings, network parameters, location characteristics, semantics, concepts e.t.c.). But several clustering approach does not adapt to the dynamic environment in which there will be continuous arrival and removal of services. As the collection of services is similar to document collection and the clustering is therefore document clustering. The existing clustering approaches are enhanced with several techniques to make suitable for dynamic nature. The clustering algorithms should include incremental procedure within it. While designing incremental system, it has to solve clustering issues, computationally efficient, cost effective. Cover coefficient is a clustering technique which cluster group of documents based on common terms and term weight. This algorithm solves important issue of finding number of clusters at initial stage. This efficient algorithm is used in incremental system suitable as C2ICM for dynamically growing database. This paper is a review about basic concepts of incremental algorithm, cover coefficient concepts used for recommendation.

Key words: Web services publishing · Web services discovery · Services discovery process and methodology

INTRODUCTION

Web service recommendation is suggesting a list of services which are very similar to search criteria. The list of services recommended are so called as neighbours to active users who are requestors. There are various techniques to extract neighbours among which collaborative filtering is a traditional one used to trace similar users. Various categories of services are available for different application areas in a registry. They differ in functional and non-functional characteristics fixed by providers.

These services are requested by the active users according to their functional and non-functional requirement with accuracy. Practically, there exists large volume of heterogeneous data. This data are frequently changed or updated in dynamic environment. Obviously, the active users are put in a situation to select a qualitative service among a long list of search result when they are not aware of these technical functional and non-functional service characteristics. There exist several recommendation systems to guide them by recommending services which are very closest to their expectation. But this recommendation system proves between them better in various aspects like speed, availability, scalability, cost e.t.c.

The system has to root out several issues like sparsity, scalability and cold start problem e.t.c. It has to prune unrelated services and services which will affect the accuracy of result. The recommendation system can be very efficient in a reduced search space with a collection of very nearest neighbours.

Grouping of similar objects is called clustering in data mining. It has been used for several applications in information retrieval. This categorization of services based on their commonalities is applied for improving

Corresponding Author: N. Kannammal, Assistant Professor, Surya Engineering College, Erode, Tamilnadu, India. E-mail: n.kannammal@yahoo.in.

computational efficiency for web mining. So, as the rate of publishing web services is growing there is a need for innovating new algorithms and techniques to be introduced in information retrieval from WWW.

Clustering is one of the data mining techniques used to categorise similar objects. This technique is very important step in recommendation algorithm which is used to confine the searching in reduced space. It uses service special characteristics to categorize them

As the services can be referred as document, the categorizing process can be called as document clustering. The document has to be efficiently arranged for effective searching [1]. Every clustering algorithm is based on hypothesis," documents relevant to a query tend to be more similar to each other than to irrelevant documents and hence are likely to be clustered together".

The whole collection of documents (services) is clustered as a pre-processing step. The documents within the cluster are more similar and are dissimilar with the document in other clusters. Some documents fall in more than one cluster forming overlapping clusters [2].

Document clusters are of either hard clusters (disjoint clusters) or soft clusters (overlapping clusters). When the document fits exactly into one cluster, then it is hard cluster. When the document fits into more than one cluster it is soft cluster [3].

There exist several basic and conventional clustering methods for document categorisation. Some more new algorithms are implemented and some are created by hybrid of basic methods. The basic clustering methods are

- Hierarchical clustering
- Partitioning clustering
- Density-based partitioning
- Grid based clustering

Other Clustering Techniques:

- Constraint-Based Clustering
- Graph Partitioning
- Clustering Algorithms and
- Supervised Learning
- Clustering algorithms in
- Machine Learning
- Scalable Clustering Algorithms
- Subspace Clustering
- Co-clustering Techniques

The paper is organized as follows. Section I introduces the basic idea of clustering. Section II details about incremental algorithm. Section III reports the related work of incremental algorithms. Section IV explains cover coefficient concept and followed by Section V details C2ICM. Section VI concludes.

Incremental Cluster Algorithm: Each clustering algorithm can categorise the document database in a way useful for browsing and searching. But as the database shrink and grow rapidly in this business environment, the clustering algorithm should be capable of working efficiently in dynamic way. And also the clustering technique should be incorporated with incremental procedure to handle addition and deletion of services (or documents) to the database. Existing clustering solution should adapt to dynamic changes. Document clustering done in a batch mode (i.e. all the documents are available at the start of clustering) is not suitable when there is continuous arrival of documents. So incremental procedure of clustering is essential when boundary of updating is not known [4]. Clustering system along with maintenance is essential for the existing dynamic environment.

There are several incremental clustering algorithm existing which differ in efficiency and effectiveness. The efficiency of any incremental algorithm is based on the idea that the new service instances should be added without disturbing existing clusters. Moreover, it should avoid the burden of reclustering. The algorithm should also solve the state when an service does not fit into any of the existing clusters [5].

General steps of Incremental Clustering Technique is

Pseudocode

Input: INS (instances), K (number of clusters),)
TH (threshold to fit instance into a cluster
Output: Clusters

1. Clusters=0
2. For All X_i □ INS
3. Set F = False
4. For All Cluster □ ClusterSet
5. If(Compare(Xi,Centroid(Cluster))= TH)
6. Include Xi to Cluster /*instance fixed
7. Update Centroid(Cluster)
8. Increment ClusterInsCount(Cluster)
9. Set F = True
10. Exit loop
11. EndIf

12. EndFor
13. If NOT(F)
14. Centroid(newCluster)=Xi /*newcluster
15. IncrementClusterInsCount(newCluster)
16. dd newCluster to ClusterSet
17. EndIf
18. EndFor

Related Work: BIRCH is a incremental algorithm that is not order dependent but suffers from working for high dimensional data [6].

COBWEB is another incremental algorithm which is sensitive to order of item. It makes use split and merge operation to add and delete new item. Based on the highest and second highest categorical utility value child nodes are removed and added [7].

CLASSIT is a derivative algorithm of COBWEB. Both the algorithm uses Normal Distribution for measuring cluster purity. Moreover, this distribution performs less for a situation called burstiness. The problem arises due to word occurrences in a document. In a word likelihood computation, a word that occur more time relatively introduces near zero factor making likelihood value of that word to zero [8].

Katz-CLASSIT follows Katz distribution for measuring cluster purity. It produces good cluster formation than the Normal distribution. Katz distribution efficiently handles the attributes which follow negative binomial distribution. Katz distribution solves burstiness problem by not only considering word occurrences but also class labels [9].

SHC is an incremental dynamic model which is based on similarity histogram clustering. Here pair wise document similarity is calculated. Also for each cluster histogram ratio is calculated. This histogram ratio is maintained by a threshold value to balance the tightness of documents to their clusters. When a new document arrives, a pre calculation is done to test whether addition decrease the histogram ratio of cluster. If so, new document is not added to that cluster [10].

Weighted co clustering algorithm that simultaneously cluster user and item. The algorithm consists of three algorithm static training, prediction, incremental training. In incremental algorithm, as the prediction is based on summary statistics of matrix average, addition of new user or item will have impact on prediction. So it should be updated. But, immediately new user or item is not fit into cluster. Instead, they are included to global transitional cluster indexed as 0. Overall matrix average is updated. During the next run of co-clustering algorithm, new user or items are fitted into their cluster. To speed up the prediction and incrementing routines, two processors were used [11].

Cover Coefficient Concept (CC): Computational complexity and cost effectiveness are important quality deciding factors for a cluster algorithm [12]. CC is a concept that is categorised under single-pass clustering methodology. Each cluster is formed on its seed document. They are called cluster seed. So, number of clusters is number of seeds. Usually, most of the clustering algorithm expect, this number of cluster value to be given as input at initial stage of clustering.

Cluster maintenance for updating clustering structure in dynamic situation is usually carried out by cluster splitting and adaptive clustering methodologies. They suffer from order dependency and at a stage performance deteriorates even for a small increase in database. Another method to maintain the cluster is by reclustering. But recomputation is very complex and does not result to linear complexity.

Clustering Process: CC result in partitioned type of clusters. It follows the following steps [13].

- Identify cluster seeds
- Cluster non-seed documents to the existing seed documents
- When non-seed documents does not match with seeds, maintain as rag-bag cluster.

According to the concept, document relationships is determined by forming cover-coefficient matrixC(mx m) from document-term matrix D(m x n).

The C matrix is formed with entry c(i,j) as

$$c_{ij} = \alpha_i x \sum_{k-1}^{n} (d_{ik} \times \beta_k \times d_{jk})$$

and

$$c_{ij}(1 \le i, j \le m)$$

Where α_i and β_k *i* th row sum and *k*th column sum respectively [13].

Every entry indicate the probability of selecting any term of d_i from d_j. This similarity measure is used as the documents that have many common terms will have high probability of being selected. Those documents appear commonly in term lists.

In C matrix,

- $0 < c_{ij} = 1$
- $c_{ii} = c_{ij}$
- $c_{i1} + c_{i2} + c_{i3} + \ldots + c_{im} = 1$

c_{ij} value indicates how much d_i is covered by d_j. Following are the inferences observed when using CC algorithm.

- Documents that have similar terms with more number of documents - c_{ii} less than 1
- Documents that have similar terms with less number of documents - c_{ii} close to 1
- Similar documents are equally covered by all other documents.
- When d_i, s document vector is a subset of $d_{j's}$, then d_j covers d_i. Also d_i cover itself to same extent.
- When two document have no common terms, their c_{ij} and c_{ji} will be zero.

Coupling and Decoupling Coefficient: When d_i is more dissimilar from other documents, c_{ii} value is high and is denoted as decoupling coefficient $\square_i = c_{ii}$. Therefore, let $\square_i$ be a coupling coefficient equal to 1- $\square_i$. Decoupling coefficient values shows the highness of difference between clusters. While the coupling coefficient shows the highness of intra cluster similarity between the documents within a cluster.

Number of Clusters: An important step of common clustering algorithms is finding number of clusters at initial stage of clustering process. Many algorithms find number of clusters randomly, user defined, using rule of thumb. This randomness affects the convergence of result point. It is faced with genetic based and partition based algorithm (e.g. PSO, KMEAN). The main strength of CC is calculating K (number of clusters) by tracing seed documents among a collection. The nearest neighbours to this seed documents are clustered together.

Based on the diagonal value of C matrix, number of cluster is found. The diagonal value c_{ii} ($\square_i$) is high when more number of dissimilar documents is in database. It is vice versa for similar documents. Here initial cluster number n_c is given as

$$n_c = \sum_{i=1}^{m} \delta_i \text{ Where } 1 = n_c = \min(m,n)$$

Cluster Seed Formation: The cluster seeds are raced out b calculating seed power to the documents. $\square_i$ (decoupling coefficient) clearly differentiate the dissimilarity between each cluster while $\square$i (coupling coefficient) gives clearly intra cluster similarity between documents inside a cluster. Using this two coefficient and D matrix with d_{ij} value, the seed power is calculated as

$$p_i = \delta_i \times \psi_i \times \sum_{j=1}^{n} d_{ij}$$

$$p_i = \delta_i \times \psi_i \times \sum_{j=1}^{n} (d_{ij} \times \delta_j' \times \psi_j')$$

The documents with high seed power is taken and used for cluster formation. The prototype characteristics are used to pull the similar documents to form clusters. The documents are listed with its seed power value.

Following are inferred to select based on term weight in document.

- When a term is having more weight but occurs very frequent or very rare in a document will not contribute to seed power value. So n_c high seed power values are taken.
- When two documents have same seed power then they are very similar documents, one is chosen arbitrarily.
- For a document to fix in a cluster, we see the seed which maximum covers it. When more than one seed covers, document is assigned to seed with high power among all of them. Also when two seed with high power situation arises, it is assigned to seed with minimum number of documents.
- Cover Coefficient Incremental Clustering Method C^2ICM

Cover coefficient algorithm is used to solve one of the issues of finding number of clusters at initial stage. This algorithm is also extended to design a dynamic incremental system. The efficient system has to update the cluster when there is addition or deletion of documents without reclustering. The extended algorithm is called C^2ICM.

The algorithm consists of following steps:

- update the database (add documents, delete documents)
- calculate *n* number of cluster for the updated database
- select *n* high seed power
- perform clustering
- repeat 1 when documents are added or deleted.

Following table show some of the existing work in which CC concept is used with other existing clustering techniques like hierarchical and partitioned. CC is a partitioned algorithm based on probability of occurrence of terms in the document [14]. From the study of existing research works, it is inferred that CC algorithm attains linear computational complexity. The algorithm is very cost effective.

Table 1: Existing works using Cover Coefficient concept for clustering

CC +	Applied to	Measures taken	Dataset --- Result obtained
k-means [15]	News portal Creation Labelling with term weighting	Compared with STC, Lingo Wfmeasure, NMI Highly effective	Ambient,ODP --- 97.3%outperforms Ambient dataset
CC summarizer [16]	Multi doc. Summarizer is created Similarity between sentence	ROUGE 1,2,3 score	N-Grams,LCS DUC2004 corpus --- 2nd rank in ROUGE score
k-means [17]	EBM (evidence based medicine) Changing number Of clusters	different measure like Euclidean, cosine, jaccard, correlation is measured	PUBMED --- Good cluster entropy value for cosine measure
Two phase ranking [18]	Reranking score is calculated Boolean ranked base result set is clustered	Precision, recall	Wikepedia2009 test bed --- Improved over baseline result

CONCLUSION

Finding number of clusters and incorporating incremental system in a clustering algorithm is essential for nowadays dynamic environment. The CC algorithm works well among all other methods of finding number of clusters. It approximates well and give good entropy value [17]. CC is efficient as it takes into account presence of term, its weight and frequency in document. Many of the research work has been proved efficienctly by hybrid of CC with other clustering algorithm. The future work will be carried out in working cover coefficient concept with remaining clustering algorithm for producing good query search result set.

REFERENCES

1. Huang, A., 2008. “Similarity measures for text document clustering,” In Proc. of the Sixth New Zealand Computer” Science Research Student Conference NZCSRSC, pp: 49-56.
2. Nicholas, O. Andrews and Edward A. Fox, 2007. “Recent developments in document clustering,” Technical report published by citeseer, pp: 1-25, Oct. 2007.
3. Chun-Ling Chen, S.C. Tseng and Tyne Liang, 2010. “An integration of Word Net and fuzzy association rule mining for multi-label document clustering,” Data and Knowledge Engineering, 69(11): 1208-1226, Nov. 2010.
4. Fazli Can and Esen Ozkarahan, 1987. ”Concepts and effectiveness of the Cover Coefficient Based Clustering Methodology for Text Databases”, Computer Science and System Analysis, Technical Report.
5. Lior Rokach and Oded Maimon, 0000. ”Clustering Methods”, Chapter 15, Department of Industrial Engineering, Tel-Aviv University.
6. Zhang, T., R. Ramakrisnan and M. Livny, 1996. “Birch: An efficient data clustering method for very large database”, pp: 103-114.

7. Douglas, H. Fisher, 1987. Knowledge acquisition via incremental conceptual clustering. Machine learning, 2: 139-172.
8. Gennari, J.H., P. Langely and D. Fisher, 1989. ModelsofIncremental concept formation. Journal of Artificial Intelligence, 40: 11-61.
9. Nachiketa Sahoo, 2006. "Incremental Hierarchical Clusterin of Text Documents", CIKM'06 Proceedings of the 15 ACM International Conference on Information Knowledge Management, pp: 357-366.
10. Khaled, M. Hammouda and Mohamed S. Kamel, 2003. "Incremental Document Clustering Using Cluster Histograms Similarity" WI, 2003 Proceedings. IEEE/WIC Inernaional Conference on Web Intelligence.
11. Thomas George and Srujana Merugu, 2005. "A Scalable Collaborative Filtering Framework based on Co- clustering", ICDM'05 Proceedings of the Fifth IEEE International Conference on Data Mining, pp: 625-628.
12. Xiaoke Su,Yang Lan, Renxia Wan and Yumin Qin, 2009. "A Fast Incremental Clustering Algorithm", Proceedings of the 2009 International Symposium Information Processing, Huangshan, P.R China.
13. Can, F., 1993. Incremental Clustering for Dynamic Processing, ACM Transaction on Information Systems, 11(2): 143-164.
14. Anil Tusel and Fazli Can, 2011. 2009. "A new approach to search result clustering and labelling", Thesis.
15. Genenc Ercan and Fazli Can, 2009. "Cover Coefficient-Based Multi-document Summarization". Springer, ECIR, pp: 670-674.
16. Sara Faisal Shah and Diego Molla, 2013. "Clustering of Medical Publications for Evidence Based Medicine Summarisation", 14th Conference on Artificial Intelligence in Medicine, AIME, Spain.
17. Kilinc, D. and A. Alpkocak, 2009. DEU at Image CLEF 2009 Wikipedia MM task: Experiments with expansion and reranking approaches. In working notes of CLEF 2009, Corfu, Greece.