Grouping with compound vision position supported relate computation

M.Krantikumar *(M.Tech Computer science & Engineering)Email: m.krantikumar@yahoo.com*
Guide name: Dr.P.Sateesh (M.Tech, PhD Associate professor)
M.V.G.R. College of engineering
Chintalavalasa, Vizianagaram dist.
www.mvgrce.edu.in

Abstract:
To make the documents in fine clustering and search for matched patterns we propose GCPSC, grouping with compound vision position supported related computation. We put all the documents in one grouped place. We take a big document, which is having large data among the available documents. Take all the prefixes and put the best possible, feasible patterns as functions. Hence these functions could be the dependency functions for the further documents discovery patterns. Time complexity is important with this scenario.

IndexTerms*: Preprocessing,Categorization, Functional dependency, Data comparison, duplication Reduction.*

Introduction:
Clustering is one of the most interesting and important topics in data mining. The aim of clustering is to find intrinsic structures in data, and organize them into meaningful subgroups for further study and analysis. There have been many clustering algorithms published every year. They can be proposed for very distinct research fields, and developed using totally different techniques and approaches. More than half a century after it was introduced, the simple algorithm k-means still remains as one of the top 10 data mining algorithms nowadays. It is the most frequently used partitioned clustering algorithm in practice. Another recent scientific discussion that k-means is the favorite algorithm that practitioners in the related fields choose to use. Needless to mention, k-means has more than a few basic drawbacks, such as sensitiveness to initialization and to cluster size, and its performance can be worse than other state-of-the-art algorithms in many domains. In spite of that, its simplicity, understandability, and scalability are the reasons for its tremendous popularity. An algorithm with adequate performance and usability in most of application scenarios could be preferable to one with better performance in some cases but limited usage due to high complexity.

While offering reasonable results, k-means is fast and easy to combine with other methods in larger systems. A common approach to the clustering problem is to treat it as an optimization process. An optimal partition is found by optimizing a particular function of similarity (or distance) among data. Basically, there is an implicit assumption that the true intrinsic structure of data could be correctly described by the similarity formula defined and embedded in the clustering criterion function. Hence, effectiveness of clustering algorithms under this approach depends on the appropriateness of the similarity measure to the data at hand. For instance, the original k-means has sum-of-squared-error objective function that uses Euclidean distance. In a very sparse and high-dimensional domain like text documents, spherical k-means, which uses cosine similarity (CS) instead of Euclidean distance as the measure, is deemed to be more suitable.

Literature survey:

Grouping is the unsubstantiated classification of patterns into groups also called clusters.

The clustering problem has been addressed in many contexts and by researchers in many disciplines; this reflects its big demand and convenience as solitary of the stepladder in probing data study. However, clustering is a not easy problem combinatorial, and differences in assumptions and contexts in different communityhavefinished the relocate of positive generic theory and methodologies measured to happen. This paper presents an indication of pattern clustering methods from analgebraic pattern respectstandpoint, with anambition of providing positiveguidance and references to essential concepts easily reached to the broad district of clustering practitioners. We present categorization of clustering techniques, and discoverbest themes and up to date advances. We also describe some important application of clustering algorithms.

In this section, we briefly review the literature related toour research in different aspects.

First, our research can be considered as performing nonredundant clustering in non-redundant clustering,we are typically given a set of data objects together withan existing clustering solution and the goal is to learn analternative clustering that captures new information aboutthe structure of the data. Existing non-redundant clusteringtechniques include the conditional information bottleneck approach, the conditional ensemble based approach and the constrained model based approach.Compared to existing non-redundant clustering techniques,the critical differences of our proposed research are:

(1) Focusing on searching for orthogonal clustering ofhigh-dimensional data, our research combines dimensionality reduction and orthogonal clustering into aunifying framework and seeks lower dimensional representation of the data that reveals non-redundant information about the data.

(2) Existing non-redundant clustering techniques are limited to finding one alternative structure given a knownStructure.

Our framework works successively to reveal a sequence of different clustering of the data. Our framework produces a set of different clustering solutions. Aclear distinction of our work from cluster ensembles is thatwe intentionally search for orthogonal clustering's and donot seek to discover consensus clustering as our end product.

seek a single final clustering solution by learning from multiple views, whereas our non-redundant multi-view clustering paradigm looks for multiple clustering solutions.An integral part of our framework is to search a clustering structure in a high-dimensional space and find the corresponding subspace that best reveals the clustering structure.In this work, after a clusteringsolution is obtained, a subspace is computed to best capture the clustering, and the clustering is then refined usingthe data projected onto the new subspace. Interestingly, ourframework works in an opposite direction. We look at a subspace that is orthogonal to the space in which the originalclustering is embedded to search for non-redundant clustering solutions.

Finally, while we search for different documents in our framework, indocument clustering, the goal is still tolearn a single clustering, where each cluster can be embedded in its own subdocuments. In contrast, our method searchesfor multiple clustering solutions, each is revealed in a different subspace.

Document categorization:

Here in this situation the duplicate documents have to be reduced**.** By comparing the documents with one by one the duplicated documents are Reduces with clustering them.

Duplication Reduction(DUP-R):

DUP-R: This is the algorithm will be used to remove the duplicate data before clustering and preprocessing, this simplifies our further clustering and also for classifications. The analysis of the grouping will be done in accordance with duplicate reduction. The reduction resembles the classifications coz while duplicating DUP-R will classify the documents in categories. So the categories will be achieved prior to further processing of the documents. Once the classification of the document is achieved, we do further processing per category. Mainly we take large data document. As the seed document and for further process we take best threshold based on document prefixes. The threshold is always is based on the categories wise, because the category is the main criteria before clustering.

While removing the duplications always we take index of all unique/duplicate items to maintain the original document copy this will be maintained but not considered in further clustering process. So the comparison is also in the form of non repetitive comparison. This always saves time complexity for duplicate reduction. If the document vector is formed with n documents then the comparison will be always less than n*n/2.Normally, regular comparison results n*n which is very time consuming process.

The Document comparison matrix is as follows: for 4 available documents:

	d1	**d2**	**d3**	**d4**
d1	$	**p1(-)**	p2(+)	p3(+)
d2	**p1(+)**	$	p4(-)	p5(-)
d3	p2(-)	p4(+)	$	p6(+)
d4	p3(-)	p5(-)	p6(-)	$

Picture: 1(*20*)

After document categorization, normally documents will be compared to reduce to less documents for marking or mining (which is of next level of data mining). In this paper we propose new comparison technique dup-r reduction for less time complexity. When ever the loop in a program with inner loop may cause duplicate comparisons, so to over come this we approach dup-r technique. This dup-r normally will reduce the comparisons repetition as in program the loop architecture will put the mark for repetition (ex: a<-->b will be marked to ignore b<→) so big comparison will be occupied with less time so that repetition will be totally vanished in the form of comparisons. And self comparison documents will be neutralized ($) as shown in (20). The above metrix will be the result for {d1, d2, d3, d4….}, so this can be extended further (future work) for *n* documents.

Document marking:

Whenever the documents are compared the documents will be put up with the marking whenever it finds the duplications. This simplifies the document clustering coz the prefix and index terms will be marked. This marking is based on comparison criteria. Once the marking is done the marked vector is updated for duplication reduction process.

Notations and algorithm for DUP-R and marking with algorithm:

Dup-R Algorithm:

Input: $\sum D$ //total no of available documents

Output: $\sum CD$ //final comparison documents for searching.

Initialization: $ג = 2mb$,Cc ← comparison condition

Count=0

$\sum P_P$←positive vector

$\sum P_n$←negitive vector

N← 0

$\sum F \leftarrow$ 0//most frequently visited terms
Loop: for each I in D
Flag ←DUP_R(i,i++)
If(flag)
P_p←(I,i++)
Else
P_n←(I,i++)
End loop

DUP_R function
Input→2 preceding documents
Output→flag with +(or)-indication
Count 1←0
Count 2←0
Count 1←FIND(fd)// first document
Count 2 ← FIND(sd)// second document.
FIND(d)
{
For each f_v in F
If(d€F)
Count 1 ++
Else
Continue
}
If(count1>count2)
flag←1
else
flag←0
end

DUP – R technique:

This is unique approach to reduce the document(s) comparisions. Normally if we have n- documents and if we start comparing the documents n^2 times the documents will be iterated for comparison. But DUP-r approach will use 2-conditions which will reduces $<n^2$ times.

{d1, d2, d3, d4, d5} are availabledocuments – **@**
Condition1: Self comparison
Condition2: repetitive comparison.

Ex: if we have **@** document set then 25times the iterations will occur for comparisions with existing approach. But DUP-r technique first reduces self comparison (*d1* with *d1* ,*d2* with *d2* etc.) this itself reduces ***n*** comparisons from total comparisons. When we take regular comparisons in the iterations if we compare *d1* with *d2* in the first iterations then that comparison will be logged with either *+ve p1* indication or *–p1* indication to ignore *d2* with *d1* that will reduce another *n/2* so total comparisions will be reduced.

Function dependency: While categorizing the data an auto generated function will be executed for categorization The documents will be categorized in 2 ways , first level category and second level category.
Attributes for first level category:

- Size according to the threshold
- Sequential / incremental data(size)
- Non special data(general or non sensitive)
- Non structured.

Attributes for second level category:

- Spatial data(non general which is sensitive)
- Structured.

(note: the document if not matching with the above categories will be saved as buffered or non processed data)

Dependency technique:
Input documents $\sum_o^n D$
Output$\sum_0^f f_l$ & $\sum_o^s S_L$
Declaration for first level function
$\lambda = 2mb$
Count=0
Flag=0
$\sum_0^b BD = 0$ //buffered document
Loop: for each d in D
Count=0
flag⇐DP_f(d)
count++

if (flag)
$F_l \leftarrow d$
Else
$BD \leftarrow d$
End loop
$F_l \Leftarrow I_N CR(F_l)$//incremented documents in size

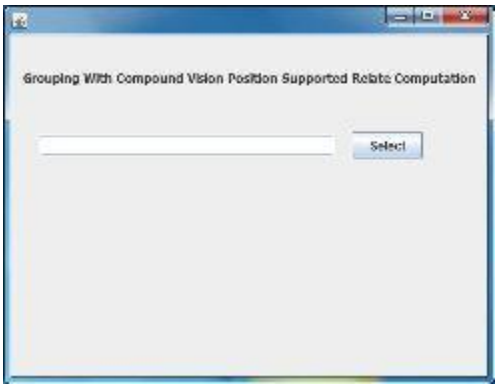
Dependency function:
$DP_f(d,ג)$
Size ⇐SIZEOF(d)
If(size>=ג)
flag⇐ *se*t
return flag
Declaration for de clustered level
ג=10
Count=0
Flog=0
Loop: for each d in D
Count =0
flog⇐ $DP_f(d)$
count ++
if(flog)
$S_l \leftarrow d$
Else
$BD \leftarrow d$
End loop

Conclusion:

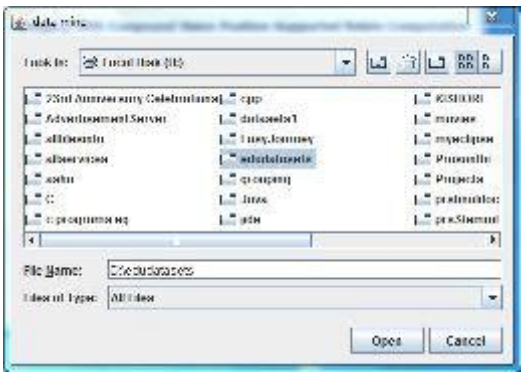
The clustering mechanism is the best way to cluster the documents in grouping them.Various mechanisms are proposed many algorithms for clustering data. Here in this paper we proposed a technique for making the documents with duplication free,By reducing the duplication of data and documents in datasets we can reduces the wastage by applying the documents cleaning technique. Documents are then clustered and forming as individual patterns from multi view point of data sets.From all these method we are making the data sets clean and feel free to the for required data effectively with low complexity.

Experimental Results:

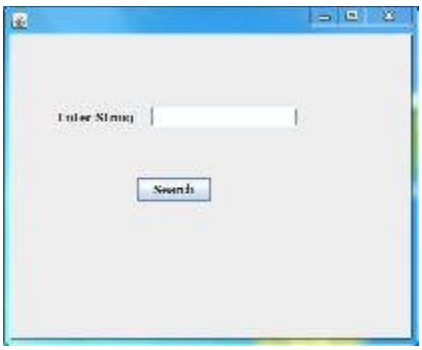
1.



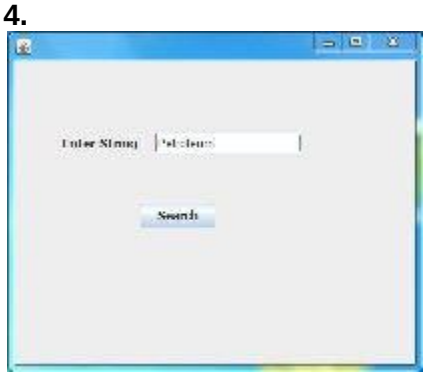
2.



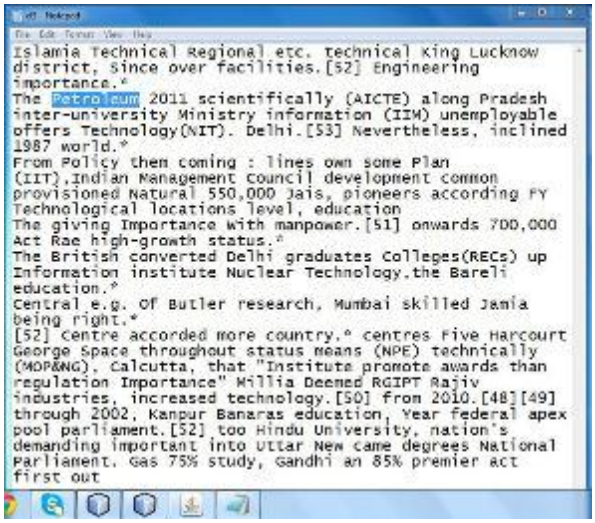
3.



4.



5.



References:

[1] W. Xu, X. Liu, and Y. Gong, "Document Clustering Based on Non- Negative Matrix Factorization," Proc. 26th Ann. Int'l ACM SIGIR Conf. Research and Development in Informaion Retrieval, pp. 267-273,2003.

[2] I.S. Dhillon, S. Mallela, and D.S. Modha, "Information-Theoretic Co-Clustering," Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 89-98, 2003.

[3] C.D. Manning, P. Raghavan, and H. Schu¨ tze, An Introduction to Information Retrieval. Cambridge Univ. Press, 2009.

[4] C. Ding, X. He, H. Zha, M. Gu, and H. Simon, "A Min-Max Cut Algorithm for Graph Partitioning and Data Clustering," Proc.IEEE Int'l Conf. Data Mining (ICDM), pp. 107-114, 2001.

[5] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Spectral Relaxation for K-Means Clustering," Proc. Neural Info. Processing Systems (NIPS), pp .1057-1064, 2001.

[6] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," IEEE Trans. Pattern Analysis Machine Intelligence, vol. 22, no. 8, pp. 888-905, Aug. 2000.

[7] I.S. Dhillon, "Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning," Proc. Seventh ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 269-274, 2001.

[8] Y. Gong and W. Xu, Machine Learning for Multimedia Content Analysis. Springer-Verlag, 2007.

[9] Y. Zhao and G. Karypis, "Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering," Machine Learning, vol. 55, no. 3, pp. 311-331, June 2004.

[10] G. Karypis, "CLUTO a Clustering Toolkit," technical report, Dept. of Computer Science, Univ. of Minnesota, http://glaros.dtc.umn. edu/~gkhome/views/cluto, 2003.