Clustering with Multi-Position based Parallel Compute

[1]V. Redya Jadav, [2]U. Varaprasad, [3]Nimishakavi Swapna
[1,2,3]Dept. of CSE, Bomma Institute of Technology and Science, Khammam, AP, India

Abstract
The major difference between a traditional dissimilarity/similarity measure and ours is that the former uses only a single viewpoint, which is the origin, while the latter utilizes many different viewpoints, which are objects assumed to not be in the same cluster with the two objects being measured. Using multiple viewpoints, more informative assessment of similarity could be achieved. Theoretical analysis and empirical study are conducted to support this claim. All clustering methods have to assume some cluster relationship among the data objects that they are applied on. Similarity between a pair of objects can be defined either explicitly or implicitly. In this paper, we introduce a novel multi-viewpoint based similarity measure and two related clustering methods. Two criterion functions for document clustering are proposed based on this new measure. We compare them with several well-known clustering algorithms that use other popular similarity measures on various document collections to verify the advantages of our proposal.

Keywords
????????? Is Missing ?????????

I. Introduction

Clustering is one of the most interesting and impor- tant 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. Nevertheless, according to a recent study , 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 partitional clustering al- gorithm in practice. Another recent scientific discussion states that k-means is the favourite algorithm that practitioners in the related fields choose to use.

Need- less 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. 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. Hence, effectiveness of clustering algorithms under this approach depends on the appropriateness of the similar- ity measure to the data at hand.

For instance, the origi- nal 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 instead of Euclidean distance as the measure, is deemed to be more suitable.

In , Banerjee et al. showed that Euclidean distance was indeed one particular form of a class of distance measures called Bregman divergences. They proposed Bregman hard-clustering algorithm, in which any kind of the Bregman divergences could be applied. Kullback- Leibler divergence was a special case of Bregman di- vergences that was said to give good clustering results on document datasets. In , Pelillo even argued that the symmetry and non-negativity as- sumption of similarity measures was actually a limitation of current state-of-the-art clustering approaches. Simultaneously, clustering still requires more robust dis- similarity or similarity measures; recent works such as illustrate this need.

The work in this paper is motivated by investigations from the above and similar research findings. It appearsto us that the nature of similarity measure plays a very important role in the success or failure of a clustering method. Our first objective is to derive a novel method for measuring similarity between data objects in sparse and high-dimensional domain, particularly text documents. We then present our proposal for document similarity measure in Section III. It is followed by two criterion functions for document clustering and their optimization algorithms in Section IV. Extensive experiments on real-world benchmark datasets are presented and discussed in Sections V and IV. Finally, conclusions and potential future work are given in Section VII.

II. Related Work

First of all, Table 1 summarizes the basic notations that will be used extensively throughout this paper to rep- resent documents and related concepts. Each document in a corpus corresponds to an m-dimensional vector d, where m is the total number of terms that the documentcorpus has. Document vectors are often subjected to some weighting schemes, such as the standard Term Frequency-Inverse Document Frequency (TF-IDF), and normalized to have unit length.. In the literature,Euclidean distance is one of the most popular measures:

$$Dist\,(d_i, d_j) = \|d_i - d_j\| \quad (1)$$

It is used in the traditional k-means algorithm. The ob- jective of k-means is to minimize the Euclidean distance between objects of a cluster and that cluster 's centroid:However, for data in a sparse and high-dimensional space, such as that in document clustering, cosine similarity is more widely used. It is also a popular similarityticularly, similarity of two document vectors d_i and d_j, $Sim(d_i, d_j)$, is defined as the cosine of the angle between them. For unit vectors, this equals to their inner product:

$Sim\,(d_i, d_j) = \cos\,(d_i, d_j) = d_i^t d_j$

Cosine measure is used in a variant of k-means called spherical k-means . While k-means aims to minimize Euclidean distance, spherical k-means intends to maximize the cosine similarity between documents in a cluster and that cluster's centroid: The major difference between Euclidean distance and cosine similarity, and therefore between k-means and spherical k-means, is that the former focuses on vector magnitudes, while the latter emphasizes on vector di- rections. Besides direct application in spherical k-means, cosine of document vectors is also widely used in many other document clustering methods as a core similarity measurement

There are many other graph partitioning methods with different cutting strategies and criterion functions, such as Average Weight and Normalized Cut, all of which have been successfully applied for document clustering using cosine as the pairwise similarity score. In , an empirical study was conducted to compare a variety of criterion functions for document clustering.

The min-max cut graph-based spectral method is an example. In graph partitioning ap- proach, document corpus is consider as a graph G = (V, E), where each document is a vertex in V and each edge in E has a weight equal to the similarity between a pair of vertices. Min-max cut algorithm tries to minimize the criterion function:

III. Multi-VI ewpoint Based Similarity

A. Our novel Similarity Measure

The cosine similarity in Eq. (3) can be expressed in the following form without changing its meaning:

$Sim(d_i, d_j) = \cos(d_i - 0, d_j - 0) = (d_i - 0)^t (d_j - 0)$

where 0 is vector 0 that represents the origin point. According to this formula, the measure takes 0 as one and only reference point. The similarity between two documents d_i and d_j is determined w.r.t. the angle between the two points when looking from the origin.

To construct a new concept of similarity, it is possible to use more than just one point of reference. We may have a more accurate assessment of how close or distant a pair of points are, if we look at them from many different viewpoints. From a third point dh , the directions and distances to di and dj are indicated respectively by the difference vectors $(d_i - d_h)$ and $(d_j - d_h)$.

As described by the above equation, similarity of two documents d_i and d_j - given that they are in the same cluster - is defined as the average of similarities measured relatively from the views of all other documents outside that cluster. What is interesting is that the simi- larity here is defined in a close relation to the clustering problem. A presumption of cluster memberships has been made prior to the measure.

The two objects to be measured must be in the same cluster, while the points from where to establish this measurement must be outside of the cluster. We call this proposal the Multi-Viewpoint based Similarity, or MVS. From this point onwards, we will denote the proposed similarity measure between two document vectors di and dj by $MVS(d_i, d_j | d_i, d_j \in S_r)$, or occasionally $MVS(d_i, d_j)$.The similarity between two points di and dj inside cluster S_r, viewed from a point d_h outside this cluster, is equal to the product of the cosine of the angle between d_i and d_j looking from d_h and the Euclidean distances from d_h to these two points. This definition is based on the assumption that dh is not in the same cluster with d_i and d_j. these distances, Eq. also provides a measure of intercluster dissimilarity, given that points d_i and d_j belong to cluster S_r, whereas dh belongs to another cluster. Hence, the effect of misleading viewpoints is constrained and reduced by the averaging step. It can be seen that this method offers more informative assessment of similarity than the single origin point based similarity measure.

B. Analysis and Practical Examples of MVS

In this section, we present analytical study to show that the proposed MVS could be a very effective similarity measure for data clustering. In order to demonstrate its advantages, MVS is compared with Cosine Similarity (CS) on how well they reflect the true group structure in document collections. Firstly, exploring Eq. (10), we have:

where $D_{S\backslash S_r} = \sum_{dh \in S\backslash S_r} d_h$ is the composite vector of all the documents outside cluster r, called the outer composite w.r.t. cluster r, and $C_{S\backslash S_r} = D_{S\backslash S_r} / (n - n_r)$ the outer centroid w.r.t. cluster r, $\forall r = 1, \ldots, k$. From Eq. (11), when comparing two pairwise similarities $MVS(d_i, d_j)$ and $MVS(d_i, d_l)$, document d_j is more similar to document d_i than the other document d_l is, if and only if:

From this condition, it is seen that even when d_l is considered "closer " to d_i in terms of CS, i.e. $\cos(d_i , d_j) \leq \cos(d_i , d_l)$, d^l can still possibly be regarded as less similar to di based on MVS if, on the contrary, it is "closer" enough to the outer centroid $C_{S\backslash Sr}$ than d_j is. This is intuitively reasonable, since the "closer " d_l is to CS\Sr , the greater the chance it actually belongs to another cluster rather than S_r and is, therefore, less similar to d_i. For this reason, MVS brings to the table an additional useful measure compared with CS.

To further justify the above proposal and analysis, we carried out a validity test for MVS and CS. The purpose of this test is to check how much a similarity measure coincides with the true class labels. It is based on one principle: if a similarity measure is appropriate for the clustering problem, for any of a document in the corpus The two datasets were preprocessed by stop-word removal and stemming. Moreover, we removed words that appear in less than two documents or more than 99.5% of the total number of documents. Finally, the documents were weighted by TF-IDF and normalized to unit vectors. The full characteristics of reuters7 and k1b are presented in fig. 3.

fig. 4, shows the validity scores of CS and MVS on the two datasets relative to the parameter percentage. The value of percentage is set at 0.001, 0.01, 0.05, 0.1, 0.2,. . . ,1.0. According to fig. 4, MVS is clearly better than that, on average, when we pick up any document and consider its neighborhood of size equal to its true class size, only 67% of that document's neighbors based on CS actually belong to its class. If based on MVS, the number of valid neighbors increases to 80%. The validity test has illustrated the potential advantage of the new multi-viewpoint based similarity measure compared to the cosine measure.

IV. Multi-VI ewpoint Based Clustering

A. Two Clustering Criterion Functions IR and IV

Having defined our similarity measure, we now formu CS for both datasets in this validity test. For example with k1b dataset at percentage = 1.0, MVS' validity score is 0.80, while that of CS is only 0.67. This indicates We would like to transform this objective function into some suitable form such that it could facilitate the optimization procedure to be performed in a simple, fast late our clustering criterion functions. The first function, called IR, is the cluster size-weighted sum of average pairwise similarities of documents in the same cluster. Firstly, let us express this sum in a general form by function FNonetheless, while the objective of min-max cut is to minimize the inverse ratio between these two terms, our aim here is to maximize their weighted differ- ence. In F^{-}, this difference term is determined for each cluster. They are weighted by the inverse of the cluster size In the formulation of IR , a cluster quality is measuredby the average pairwise similarity between documentsto sensitiveness to the size and tightness of the clusters. With CS, for example, pairwise similarity of documents in a sparse cluster is usually smaller than those in a The clustering process terminates when an iteration completes without any documents being moved to new clusters.

```
1: procedure INITIALIZATION
2:   Select k seeds s_1, ..., s_k randomly
3:   cluster[d_i] ← p = arg max_r {s_r^t d_i}, ∀i = 1, ..., n
4:
5: end procedure
6: procedure REFINEMENT
7:   repeat
8:     {v[1 : n]} ← random permutation of {1, ..., n}
9:     for j ← 1 : n do
10:      i ← v[j]
11:      p ← cluster[d_i]
12:      ΔI_p ← I(n_p − 1, D_p − d_i) − I(n_p, D_p)
13:      q ← arg max_{r≠p} I(n_r+1, D_r+d_i) − I(n_r, D_r)
14:      ΔI_q ← I(n_q + 1, D_q + d_i) − I(n_q, D_q)
15:      if ΔI_p + ΔI_q > 0 then
16:        Move d_i to cluster q: cluster[d_i] ← q
17:        Update D_p, n_p, D_q, n_q
18:      end if
19:    end for
20:  until No move for all n documents
21: end procedure
```

Fig. 5: Algorithm: Incremental Clustering

intracluster similarity measure and an inter-cluster similarity measure, respectively. The first term is actually equivalent to an element of the sum in spherical k-means objective function in Eq. (4); the second one is similar to an element of the sum in min-max cut criterion in Eq. (6), but with D_r as scaling factor instead of D_r 2. We have presented our clustering criterion functions IR and IV in the simple forms. Next, we show how to perform clustering by using a greedy algorithm to optimize these functions.

4.2 Optimization algorithm and complexity

We denote our clustering framework by MVSC, meaning Clustering with Multi-Viewpoint based Similarity. Sub- sequently, we have MVSC-IR and MVSC-IV , which are MVSC with criterion function IR and IV respectively. The main goal is to perform document clustering by opti- mizing IR in Eq. (16) and IV in Eq. (18). For this purpose, the incremental k-way algorithm [18], [29] - a sequential version of k-means - is employed. Considering that the expression of IV in Eq. (18) depends only on nr and Dr , r = 1,..., k, IV can be written in a general form

$$I_V = \sum_{r=1}^{k} I_r(n_r, D_r) \qquad (19)$$

where I_r (n_r , D_r) corresponds to the objective value of cluster r. The same is applied to IR . With this general form, the incremental optimization algorithm, which has two major steps Initialization and Refinement, is described in fig. 5. At Initialization, k arbitrary doc- uments are selected to be the seeds from which intial partitions are formed. Refinement is a procedure that consists of a number of iterations. During each iteration, the n documents are visited one by one in a totally random order. Each document is checked if its move to another cluster results in improvement of the objective function. If yes, the document is moved to the cluster that leads to the highest improvement. If no clusters are better than the current cluster, the document is not moved. The clustering process terminates when an iteration completes without any documents being moved to new clusters. Unlike the traditional k-means, this algorithm is a stepwise optimal procedure. While k-means only updates after all n documents have been reassigned, the incremental clustering algorithm updates immediately whenever each document is moved to new cluster. Since every move when happens increases the objective function value, convergence to a local optimum is guaranteed.

During the optimization procedure, in each iteration, the main sources of computational cost are:

- Searching for optimum clusters to move individual documents to: O(nz • k).
- Updating composite vectors as a result of such moves: O(m • k).

where n_z is the total number of non-zero entries in all document vectors. Our clustering approach is partitional and incremental; therefore, computing similarity matrix is absolutely not needed. If τ denotes the number of iterations the algorithm takes, since n_z is often several tens times larger than m for document domain, the computational complexity required for clustering with I_R and I_V is O(nz • k • τ).

V. Performance Evaluation of MVSC

To verify the advantages of our proposed methods, we evaluate their performance in experiments on document data. The objective of this section is to compare MVSC- I_R and MVSC-I_V with the existing algorithms that also use specific similarity measures and criterion functions for document clustering. The similarity measures to be compared includes Euclidean distance, cosine similarity and extended Jaccard coefficient.

A. Document Collections

The data corpora that we used for experiments consist of twenty benchmark document datasets. Besides reuters7 and k1b, which have been described in details earlier, we included another eighteen text collections so that the examination of the clustering methods is more thor ough and exhaustive. Similar to k1b, these datasets are provided together with CLUTO by the toolkit's authors [19]. They had been used for experimental testing in previous papers, and their source and origin had also been described in details [30], [31]. Table 2 summarizes their characteristics. The corpora present a diversity of size, number of classes and class balance. They were all preprocessed by standard procedures, including stop- word removal, stemming, removal of too rare as well as

Table 2: Document datasets

Data	Source	c	n	m	Balance
fbis	TREC	17	2,463	2,000	0.075
hitech	TREC	6	2,301	13,170	0.192
k1a	WebACE	20	2,340	13,859	0.018
k1b	WebACE	6	2,340	13,859	0.043
la1	TREC	6	3,204	17,273	0.290
la2	TREC	6	3,075	15,211	0.274
re0	Reuters	13	1,504	2,886	0.018
re1	Reuters	25	1,657	3,758	0.027
tr31	TREC	7	927	10,127	0.006
reviews	TREC	5	4,069	23,220	0.099
wap	WebACE	20	1,560	8,440	0.015
classic	CACM/CISI/ CRAN/MED	4	7.089	12.009	0.323
la12	TREC	6	6,279	21,604	0.282
new3	TREC	44	9,558	36,306	0.149
sports	TREC	7	8,580	18,324	0.036
tr11	TREC	9	414	6,424	0.045
tr12	TREC	8	313	5,799	0.097
tr23	TREC	6	204	5,831	0.066
tr45	TREC	10	690	8,260	0.088
reuters7	Reuters	7	2,500	4,977	0.082

c: # of classes, n: # of documents, m: # of words
Balance= (smallest class size)/(largest class size)

too frequent words, TF-IDF weighting and normaliza.

B. Experimental Setup and Evaluation

To demonstrate how well MVSCs can perform, we compare them with five other clustering methods on the twenty datasets in Table 2. In summary, the seven clustering algorithms are:

- MVSC-IR: MVSC using criterion function I_R
- MVSC-IV : MVSC using criterion function I_V
- k-means: standard k-means with Euclidean distance
- Spkmeans: spherical k-means with C_S
- graphCS: CLUTO's graph method with C_S
- graphEJ: CLUTO's graph with extended Jaccard
- MMC: Spectral Min-Max Cut algorithm [13]

Our MVSC-IR and MVSC-IV programs are implemented in Java. The regulating factor α in IR is always set at 0.3 during the experiments. We observed that this is one of the most appropriate values. A study on MVSC-IR's performance relative to different α values is presented in a later section. The other algorithms are provided by the C library interface which is available freely with the CLUTO toolkit [19]. For each dataset, cluster number is predefined equal to the number of true class, i.e. k = c. None of the above algorithms are guaranteed to find global optimum, and all of them are initialization- dependent. Hence, for each method, we performed clus- tering a few times with randomly initialized values, and chose the best trial in terms of the corresponding objective function value. In all the experiments, each test run consisted of 10 trials. Moreover, the result reported here on each dataset by a particular clustering method is the average of 10 test runs.

After a test run, clustering solution is evaluated by comparing the documents' assigned labels with their true labels provided by the corpus. Three types of ex- ternal evaluation metric are used to assess clustering performance. They are the FScore, Normalized Mutual Information (NMI) and Accuracy. FScore is an equally weighted combination of the "precision" (P) and "re- call" (R) values used in information retrieval. Given a clustering solution, FScore is determined as:

$$FScore = \sum_{i=1}^{k} \frac{n_i}{n} \max_j (F_{i,j})$$

$$\text{where } F_{i,j} = \frac{2 \times P_{i,j} \times R_{i,j}}{P_{i,j} + R_{i,j}}; P_{i,j} = \frac{n_{i,j}}{n_j}, R_{i,j} = \frac{n_{i,j}}{n_i}$$

where n_i denotes the number of documents in class i, n_j the number of documents assigned to cluster j, and n_{ij} the number of documents shared by class i and cluster j. From another aspect, NMI measures the information the true class partition and the cluster assignment share. It measures how much knowing about the clusters helps us know about the classes:

$$NMI = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} n_{i,j} \log \frac{n \cdot n_{i,j}}{n_i \cdot n_j}}{\sum_{i=1}^{k} n_i \log \frac{n_i}{n} \sum_{j=1}^{k} n_j \log \frac{n_j}{n}}$$

Finally, Accuracy measures the fraction of documents that are correctly labels, assuming a one-to-one correspon- dence between true classes and assigned clusters. Let q denote any possible permutation of index set {1,... , k}, Accuracy is calculated by:

$$Accuracy = \frac{1}{n} \max_q \sum_{i=1}^{k} n_{i,q(i)}$$

The best mapping q to determine Accuracy could be found by the Hungarian algorithm2 . For all three metrics, their range is from 0 to 1, and a greater value indicates a better clustering solution.

C. Results

Fig. 6 shows the Accuracy of the seven clustering al- gorithms on the twenty text collections. Presented in a different way, clustering results based on FScore and NMI are reported in Table 3 and Table 4 respectively. For each dataset in a row, the value in bold and underlined is the best result, while the value in bold only is the second to best.

It can be observed that MVSC-IR and MVSC-IV per- form consistently well. In Fig. 6, 19 out of 20 datasets, except reviews, either both or one of MVSC approaches are in the top two algorithms. The next consistent per- former is Spkmeans. The other algorithms might work well on certain dataset. For example, graphEJ yields outstanding result on classic; graphCS and MMC are

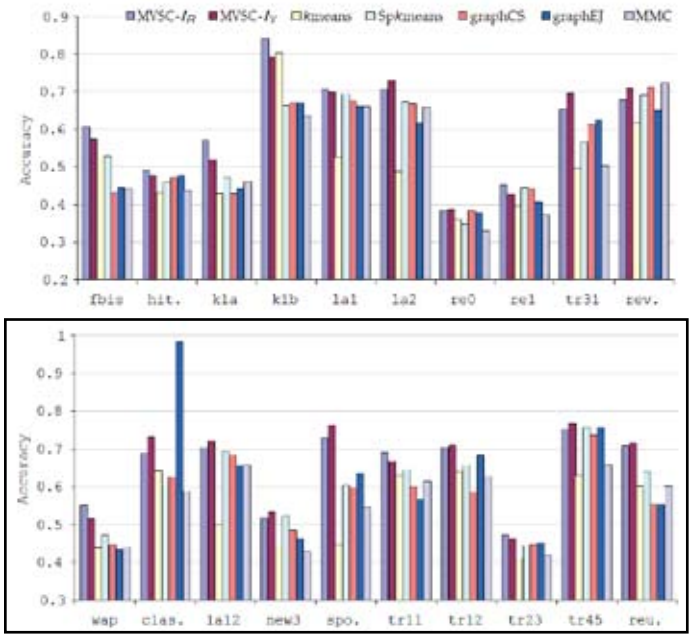

Fig. 6. Clustering results in Accuracy. Left-to-right in legend corresponds to left-to-right in the plot

good on reviews. But they do not fare very well on the rest of the collections.

To have a statistical justification of the clustering performance comparisons, we also carried out statistical significance tests. Each of MVSC-I_R and MVSC-I_V was paired up with one of the remaining algorithms for a paired t-test [32]. Given two paired sets X and Y of N measured values, the null hypothesis of the test is that the differences between X and Y come from a population with mean 0. The alternative hypothesis is that the paired sets differ from each other in a significant way. In our experiment, these tests were done based on the evaluation values obtained on the twenty datasets. The typical 5% significance level was used. For example, considering the pair (MVSC-I_R, k-means), from Table 3, it is seen that MVSC-I_R dominates k-means w.r.t. FScore. If the paired t-test returns a p-value smaller than 0.05, we reject the null hypothesis and say that the dominance is significant. Otherwise, the null hypothesis is true and the comparison is considered insignificant.

The outcomes of the paired t-tests are presented in Table 5. As the paired t-tests show, the advantage of MVSC-I_R and MVSC-I_V over the other methods is statistically significant. A special case is the

graphEJ algorithm. On the one hand, MVSC-I_R is not significantly better than graphEJ if based on FScore or NMI. On the other hand, when MVSC-I_R and MVSC-I_V are tested obviously bet- ter than graphEJ, the p-values can still be considered relatively large, although they are smaller than 0.05. The reason is that, as observed before, graphEJ's results on classic dataset are very different from those of the other algorithms. While interesting, these values can be con- sidered as outliers, and including them in the statistical tests would affect the outcomes greatly. Hence, we also report in Table 5 the tests where classic was excluded and only results on the other 19 datasets were used

Table 3: Clustering results in FScore

Data	MVSC-I_R	MVSC-I_V	k-means	Spkmeans	graphCS	graphEJ	MMC
fbis	.645	.613	.578	.584	.482	.531	.556
hitech	.512	.522	.467	.494	.472	.437	.468
k1a	.620	.592	.502	.545	.492	.527	.524
k1b	.873	.775	.825	.729	.740	.743	.707
la1	.719	.723	.565	.719	.689	.679	.693
la2	.721	.749	.538	.703	.689	.633	.698
re0	.460	.458	.421	.421	.468	.454	.390
re1	.514	.492	.456	.499	.487	.457	.443
tr31	.728	.780	.585	.679	.689	.698	.607
reviews	.734	.748	.644	.730	.759	.690	.749
wap	.610	.571	.516	.545	.513	.497	.513
classic	.658	.734	.713	.687	.708	.983	.657
la12	.719	.735	.559	.722	.706	.671	.693
new3	.548	.547	.500	.558	.535	.496	.482
sports	.803	.804	.499	.702	.689	.696	.650
tr11	.749	.728	.705	.719	.665	.658	.695
tr12	.743	.758	.699	.715	.642	.722	.700
tr23	.560	.553	.486	.523	.522	.531	.485
tr45	.787	.788	.692	.799	.778	.798	.720
reuters7	.774	.775	.658	.718	.651	.670	.687

Table 4: Clustering Results in NMI

Data	MVSC-I_R	MVSC-I_V	k-means	Spkmeans	graphCS	graphEJ	MMC
fbis	.606	.598	.584	.593	.527	.524	.556
hitech	.323	.329	.270	.298	.279	.292	.283
k1a	.612	.594	.563	.596	.537	.571	.588
k1b	.739	.652	.629	.649	.635	.650	.645
la1	.569	.571	.397	.565	.490	.485	.553
la2	.568	.590	.381	.563	.496	.478	.566
re0	.399	.402	.388	.399	.367	.342	.414
re1	.591	.583	.532	.593	.581	.566	.515
tr31	.613	.658	.488	.594	.577	.580	.548
reviews	.584	.603	.460	.607	.570	.528	.639
wap	.611	.585	.568	.596	.557	.555	.575
classic	.574	.644	.579	.577	.558	.928	.543
la12	.574	.584	.378	.568	.496	.482	.558
new3	.621	.622	.578	.626	.580	.580	.577
sports	.669	.701	.445	.633	.578	.581	.591
tr11	.712	.674	.660	.671	.634	.594	.666
tr12	.686	.686	.647	.654	.578	.626	.640
tr23	.432	.434	.363	.413	.344	.380	.369
tr45	.734	.733	.640	.748	.726	.713	.667
reuters7	.633	.632	.512	.612	.503	.520	.591

Under this circumstance, both MVSC-I_R and MVSC-I_V out perform graphEJ significantly with good p-values.

D. Effect of α on MVSC-IR 's performance

It has been known that criterion function based partitional clustering methods can be sensitive to cluster size and balance. In the formulation of IR in Eq. (16), there exists parameter α which is called the regulating factor, α ∈ [0, 1]. To examine how the determination of α could affect MVSC-IR's performance, we evaluated MVSC-I_R with different values of α from 0 to 1, with 0.1 incremental interval. The assessment was done based on the clustering results in NMI, FScore and Accuracy, each averaged over all the twenty given datasets. Since the evaluation metrics for different datasets could be very different from each other, simply taking the average over all the datasets would not be very meaningful. Hence, we employed the method used in [18] to transform the metrics into relative metrics before averaging. On a particular document collection S, the relative FScore

Table 5: Statistical significance of comparisons based on paired t-tests with 5% significance level

		k-means	Spkmeans	graphCS	graphEJ*	MMC
FScore	MVSC-I_R	>>	>>	>>	> (>>)	>>
		1.77E-5	1.60E-3	4.61E-4	.056 (7.68E-6)	3.27E-6
	MVSC-I_V	>>	>>	>>	>> (>>)	>>
		7.52E-5	1.42E-4	3.27E-5	.022 (1.50E-6)	2.16E-7
NMI	MVSC-I_R	>>	>>	>>	> (>>)	>>
		7.42E-6	.013	2.39E-7	.060 (1.65E-8)	8.72E-5
	MVSC-I_V	>>	>>	>>	>> (>>)	>>
		4.27E-5	.013	4.07E-7	.029 (4.36E-7)	2.52E-4
Accuracy	MVSC-I_R	>>	>>	>>	>> (>>)	>>
		1.45E-6	1.50E-4	1.33E-4	.028 (3.29E-5)	8.33E-7
	MVSC-I_V	>>	>>	>>	>> (>>)	>>
		1.74E-5	1.82E-4	4.19E-5	.014 (8.61E-6)	9.80E-7

">>" (or "C") indicates the algorithm in the row performs significantly better (or worse) than the one in the column; ">" (or "<") indicates an insignificant comparison. The values right below the symbols are p-values of the t-tests.

* Column of graphEJ: entries in parentheses are statistics when classic dataset is not included.

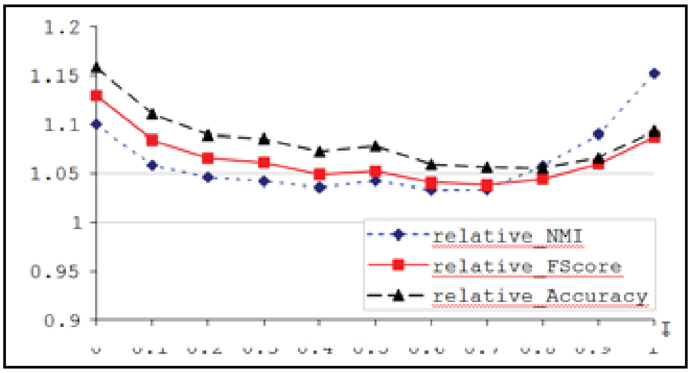

Fig. 7. MVSC-I_R 's performance with respect to α.

measure of MVSC-I_R with $\alpha = \alpha_i$ is determined as following:

$$relative\_FScore(I_R; S, \alpha_i) = \frac{\max_{\alpha_j} \{FScore(I_R; S, \alpha_j)\}}{FScore(I_R; S, \alpha_i)}$$

where αi , αj ∈ {0.0, 0.1,... , 1.0}, FScore(IR ; S, αi) is the FScore result on dataset S obtained by MVSC-IR with $\alpha = \alpha_i$. The same transformation was applied to NMI and Accuracy to yield relative NMI and relative Accuracy respectively. MVSC-I_R performs the best with an αi if its relative measure has a value of 1. Otherwise its relative measure is greater than 1; the larger this value is, the worse MVSC-IR with αi performs in comparison with other settings of α. Finally, the average relative measures were calculated over all the datasets to present the overall performance.

Fig. 7, shows the plot of average relative FScore, NMI and Accuracy w.r.t. different values of α. In a broad view, MVSC-IR performs the worst at the extreme values of α (0 and 1), and tends to get better when α is set at some soft values in between 0 and 1. Based on our experimental study, MVSC-IR always produces results within 5% of the best case, regarding any types of evaluation metric, with α from 0.2 to 0.8.

VI. MVSC as Refinement for k-MEANS

From the analysis of Eq. (12) in Section 3.2, MVS pro- vides an additional criterion for measuring the simi- larity among documents compared with CS. Alterna- tively, MVS can be considered as a refinement for CS, and consequently MVSC algorithms as refinements for spherical k-means, which uses CS. To further investigate the appropriateness and effectiveness of MVS and its

clustering algorithms, we carried out another set of experiments in which solutions obtained by Spkmeans were further optimized by MVSC-IR and MVSC-IV . The rationale for doing so is that if the final solutions by MVSC-IR and MVSC-IV are better than the intermediate ones obtained by Spkmeans, MVS is indeed good for the clustering problem. These experiments would reveal more clearly if MVS actually improves the clustering performance compared with CS.

In the previous section, MVSC algorithms have been compared against the existing algorithms that are closely related to them, i.e. ones that also employ similarity measures and criterion functions. In this section, we make use of the extended experiments to further compare the MVSC with a different type of clustering approach, the NMF methods [10], which do not use any form of explicitly defined similarity measure for documents.

A. TDT2 and Reuters-21578 Collections

For variety and thoroughness, in this empirical study, we used two new document copora described in Table 6: TDT2 and Reuters-21578. The original TDT2 corpus3 , which consists of 11,201 documents in 96 topics (i.e. classes), has been one of the most standard sets for document clustering purpose. We used a sub-collection of this corpus which contains 10,021 documents in the largest 56 topics. The Reuters-21578 Distribution 1.0 has been mentioned earlier in this paper. The original corpus consists of 21,578 documents in 135 topics. We used a

Table 6: TDT2 and Reuters-21578 Document Corpora

	TDT2	*Reuters-21578*
Total number of documents	10,021	8,213
Total number of classes	56	41
Largest class size	1,844	3,713
Smallest class size	10	10

sub-collection having 8,213 documents from the largest 41 topics. The same two document collections had been used in the paper of the NMF methods [10]. Documents that appear in two or more topics were removed, and the remaining documents were preprocessed in the same way as in Section 5.1.

B. Experiments and Results

The following clustering methods:

- Spkmeans: spherical k-means
- rMVSC-IR: refinement of Spkmeans by MVSC-IR
- rMVSC-IV : refinement of Spkmeans by MVSC-IV
- MVSC-IR: normal MVSC using criterion IR
- MVSC-IV : normal MVSC using criterion IV

and two new document clustering approaches that do not use any particular form of similarity measure:

- NMF: Non-negative Matrix Factorization method
- NMF-NCW: Normalized Cut Weighted NMF

were involved in the performance comparison. When used as a refinement for Spkmeans, the algorithms rMVSC-IR and rMVSC-IV worked directly on the output solution of Spkmeans. The cluster assignment produced by Spkmeans was used as initialization for both rMVSC-IR and rMVSC-IV. We also investigated the performance of the original MVSC-IR and MVSC-IV further on the new datasets. Besides, it would be interesting to see how they and their Spkmeans-initialized versions fare against each other. What is more, two well-known document clustering approaches based on non-negative matrix factorization, NMF and NMF-NCW [10], are also included for a comparison with our algorithms, which use explicit MVS measure.

During the experiments, each of the two corpora in Table 6 were used to create 6 different test cases, each of which corresponded to a distinct number of topics used (c = 5,... , 10). For each test case, c topics were randomly selected from the corpus and their documents were mixed together to form a test set. This selection was repeated 50 times so that each test case had 50 different test sets. The average performance of the clustering algorithms with k = c were calculated over these 50 test sets. This experimental set-up is inspired by the similar experiments conducted in the NMF paper [10]. Furthermore, similar to previous experimental setup in Section 5.2, each algorithm (including NMF and NMF-NCW) actually considered 10 trials on any test set before using the solution of the best obtainable objective func- tion value as its final output. The clustering results on TDT2 and Reuters-21578 are shown in Table 7 and 8 respectively. For each test case in

Table 7: Clustering Results on TDT2

	NMI						*Accuracy*					
Algorithms	k=5	k=6	k=7	k=8	k=9	k=10	k=5	k=6	k=7	k=8	k=9	k=10
Spkmeans	.690	.701	.700	.677	.681	.676	.708	.689	.668	.620	.605	.578
rMVSC-I_R	.758	.777	.766	.749	.738	.699	.855	.846	.822	.802	.760	.722
rMVSC-I_V	.740	.764	.742	.729	.718	.676	.839	.837	.801	.785	.736	.701
MVSC-I_R	.719	**.790**	**.757**	**.760**	**.764**	**.722**	**.884**	**.867**	**.875**	**.840**	**.832**	**.780**
MVSC-I_V	.775	.785	.779	.745	.755	.714	**.886**	**.871**	.870	.825	.818	.777
NMF	.621	.630	.607	.581	.593	.555	.697	.686	.642	.604	.578	.555
NMF-NCW	.713	.746	.723	.707	.702	.659	.788	.821	.764	.749	.725	.675

Table 8: Clustering Results on Reuters-21578

	NMI						*Accuracy*					
Algorithms	k=5	k=6	k=7	k=8	k=9	k=10	k=5	k=6	k=7	k=8	k=9	k=10
Spkmeans	.370	.435	.389	.336	.348	.428	.512	.509	.454	.390	.380	.429
rMVSC-I_R	**.386**	**.448**	.406	.347	.359	.433	**.591**	**.592**	.522	.445	.437	.485
rMVSC-I_V	**.395**	.439	.409	.351	**.361**	.434	.591	.573	.529	.453	.448	.477
MVSC-I_R	.377	.442	**.418**	**.354**	.356	**.441**	.582	**.588**	.538	**.473**	**.477**	.505
MVSC-I_V	.375	**.444**	**.414**	**.357**	**.360**	**.438**	.580	**.588**	**.552**	**.473**	**.452**	**.512**
NMF	.321	.369	.341	.289	.278	.359	.553	.534	.479	.423	.388	.430
NMF-NCW	.355	.413	.387	.341	.344	.413	.608	.580	.535	.466	.432	.493

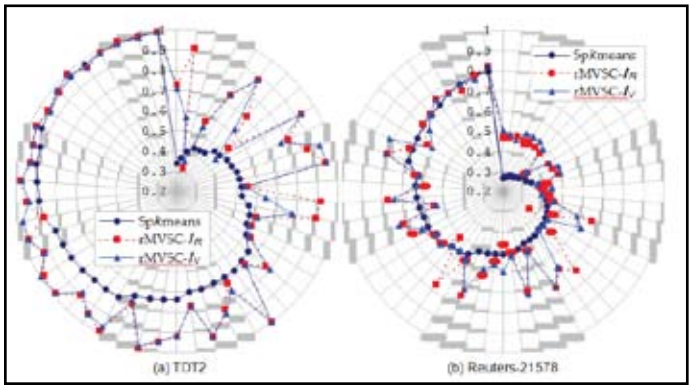

Fig. 8: Accuracies on the 50 test sets (in sorted order of Spkmeans) in the test case k = 5.

a column, the value in bold and underlined is the best among the results returned by the algorithms, while the value in bold only is the second to best. From the tables, several observations can be made. Firstly, MVSC-IR and MVSC-IV continue to show they are good clustering algorithms by outperforming other methods frequently. They are always the best in every test case of TDT2. Compared with NMF-NCW, they are better in almost all the cases, except only the case of Reuters-21578, k = 5, where NMF-NCW is the best based on Accuracy.

The second observation, which is also the main objec- tive of this empirical study, is that by applying MVSC to refine the output of spherical k-means, clustering solutions are improved significantly. Both rMVSC-IR and rMVSC-IV lead to higher NMIs and Accuracies than Spkmeans in all the cases. Interestingly, there are many circumstances where Spkmeans' result is worse than that of

NMF clustering methods, but after refined by MVSCs, it becomes better. To have a more descriptive picture of the improvements, we could refer to the radar charts in Fig. 8. The figure shows details of a particular test case where k = 5. Remember that a test case consists of 50 dif- ferent test sets. The charts display result on each test set, including the accuracy result obtained by Spkmeans, and the results after refinement by MVSC, namely rMVSC- IR and rMVSC-IV . For effective visualization, they are sorted in ascending order of the accuracies by Spkmeans (clockwise). As the patterns in both Fig. 8(a) and Fig. 8(b) reveal, improvement in accuracy is most likely attainable by rMVSC-IR and rMVSC-IV . Many of the improve- ments are with a considerably large margin, especially when the original accuracy obtained by Spkmeans is low.

There are only few exceptions where after refinement, accuracy becomes worst. Nevertheless, the decreases in such cases are small.

Finally, it is also interesting to notice from Table 7 and Table 8 that MVSC preceded by spherical k-means does not necessarily yields better clustering results than MVSC with random initialization. There are only a small number of cases in the two tables that rMVSC can be found better than MVSC. This phenomenon, however, is understandable. Given a local optimal solution returned by spherical k-means, rMVSC algorithms as a refinement method would be constrained by this local optimum itself and, hence, their search space might be restricted. The original MVSC algorithms, on the other hand, are not subjected to this constraint, and are able to follow the search trajectory of their objective function from the beginning. Hence, while performance improvement after refining spherical k-means' result by MVSC proves the appropriateness of MVS and its criterion functions for document clustering, this observation in fact only reaffirms its potential.

VII. Conclusions and Future Work

In this paper, Based on MVS, two criterion functions, IR and IV , and their respective clustering algorithms, MVSC-IR and MVSC-IV , have been intro- duced. Compared with other state-of-the-art clustering methods that use different types of similarity measure, on a large number of document datasets and under different evaluation metrics, the proposed algorithms show that they could provide significantly improved clustering performance. we propose a Multi-Viewpoint based Similarity measuring method, named MVS. Theoretical analysis and empirical examples show that MVS is potentially more suitable for text documents than the popular cosine similarity.

The key contribution of this paper is the fundamental concept of similarity measure from multiple viewpoints. Future methods could make use of the same principle, but define alternative forms for the relative similarity in Eq. (10), or do not use average but have other methods to combine the relative similarities according to the differ- ent viewpoints. Besides, this paper focuses on partitional clustering of documents. In the future, it would also be possible to apply the proposed criterion functions for hi- erarchical clustering algorithms. Finally, we have shown the application of MVS and its clustering algorithms for text data. It would be interesting to explore how they work on other types of sparse and high-dimensional data.

References

[1] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, D. Steinberg,"Top 10 algorithms in data mining", Knowl. Inf. Syst., Vol. 14, No. 1, pp. 1–37, 2007.

[2] I. Guyon, U. von Luxburg, R. C. Williamson,"Clustering: Science or Art?", NIPS'09 Workshop on Clustering Theory, 2009.

[3] I. Dhillon, D. Modha,"Concept decompositions for large sparse text data using clustering", Mach. Learn., Vol. 42, No. 1-2, pp. 143–175, Jan 2001.

[4] S. Zhong,"Efficient online spherical K-means clustering", in IEEE IJCNN, 2005, pp. 3180–3185.

[5] A. Banerjee, S. Merugu, I. Dhillon, J. Ghosh,"Clustering with Bregman divergences," J. Mach. Learn. Res., Vol. 6, pp. 1705–1749, Oct 2005.

[6] E. Pekalska, A. Harol, R. P. W. Duin, B. Spillmann, H. Bunke,"Non-Euclidean or non-metric measures can be informative", in Structural, Syntactic, and Statistical Pattern Recognition, ser. LNCS, Vol. 4109, 2006, pp. 871–880.

[7] M. Pelillo,"What is a cluster? Perspectives from game theory", in Proc. of the NIPS Workshop on Clustering Theory, 2009.

[8] D. Lee, J. Lee,"Dynamic dissimilarity measure for support based clustering", IEEE Trans. on Knowl. and Data Eng., Vol. 22, No. 6, pp. 900–905, 2010.

[9] A. Banerjee, I. Dhillon, J. Ghosh, S. Sra,"Clustering on the unit hypersphere using von Mises-Fisher distributions", J. Mach. Learn. Res., Vol. 6, pp. 1345–1382, Sep 2005.

[10] W. Xu, X. Liu, Y. Gong,"Document clustering based on non negative matrix factorization", in SIGIR, 2003, pp. 267–273.

[11] I. S. Dhillon, S. Mallela, D. S. Modha,"Information-theoretic co-clustering", in KDD, 2003, pp. 89–98.

[12] C. D. Manning, P. Raghavan, H. Schu¨ tze,"An Introduction to Information Retrieval", Press, Cambridge U., 2009.

[13] C. Ding, X. He, H. Zha, M. Gu, H. Simon,"A min-max cut algorithm for graph partitioning and data clustering", in IEEE ICDM, 2001, pp. 107–114.

V. Redya Jadav M.Tech(cs), Head of the Department Department of CSE, I am having 12 years of experience in teaching Computer Technologies published various research articles in National and International Journals.

I N.Swapna, i completed M.Sc(CS) in Kakatiya University on 2010, Pursuing M.Tech(C.S.E) in Bomma institute of technology And Science,JNTU, Hyderabad.

U.Varaprasad , completed B.Tech in Nagarjuna University on 2010 and M.Tech in JNTU, Hyderabad in 2010. And i working as a Associate Professor in Bomma Institute of Technology and Science, Khammam.