V.Leela Prasad, B.Simmi Cintre / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 4, July-August 2012, pp.409-420

Analysis of Novel Multi-Viewpoint Similarity Measures

V.Leela Prasad*, B.Simmi Cintre**
*(Student Scholar (MTech), Department of CSE, Adams Engineering College, JNTUH, Khammam,AP-507115, India)
** (Associate professor, Department of CSE, Adams Engineering College, JNTUH, Khammam, AP-507115, India)

Abstract—

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. 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. 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.

Key Terms—Document clustering, text mining, similarity measure, Clustering methods

I. 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. Nevertheless, according to a recent study [1], 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 algorithm in practice. Another recent scientific discussion [2] states that k-means is the favourite 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.

Our study of similarity of clustering was initially motivated by a research on automated text categorization of foreign language texts, as explained below. As the amount of digital documents has been increasing dramatically over the years as the Internet grows, information management, search, and retrieval, etc., have become practically important problems. Developing methods to organize large amounts of unstructured text documents into a smaller number of meaningful clusters would be very helpful as document clustering is vital to such tasks as indexing, filtering, automated metadata generation, word sense disambiguation, population of hierarchical catalogues of web resources and, in general, any application requiring document organization .

Document clustering is also useful for topics such as Gene Ontology in biomedicine where hierarchical catalogues are needed. To deal with the large amounts of data, machine learning approaches have been applied to perform Automated Text Clustering (ATC). Given an unlabeled dataset, this ATC system builds clusters of documents that are hopefully similar to clustering (classification, categorization, or labeling) performed by human experts. To identify a suitable tool and algorithm for clustering that produces the best clustering solutions, it becomes necessary to have a method for comparing the results of different clustering algorithms. Though considerable work has been done in designing clustering algorithms, not much research has been done on formulating a measure for the similarity of two different clustering algorithms. Thus, the main goal of this paper is to: First, propose an algorithm for performing similarity analysis among different clustering algorithms; second, apply the algorithm to calculate similarity of various pairs of clustering methods applied to a Portuguese corpus and the Iris dataset; finally, to cross validate the results of similarity analysis with the Euclidean (centroids) distances and Pearson correlation coefficient, using the same datasets. Possible applications are discussed.

The work in this paper is motivated by investigations from the above and similar research findings. It appears to 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. From the proposed similarity measure, we then formulate new clustering criterion functions and introduce their respective clustering algorithms, which are fast and scalable like k-means, but are also capable of providing high-quality and consistent performance.

The remaining of this paper is organized as follows. In Section 2, we review related literature on similarity and clustering of documents. We then present our proposal for document similarity measure in Section 3.It is followed by two criterion functions for document clustering and their optimization algorithms in Section 4. Extensive experiments on real-world benchmark datasets are presented and discussed in Sections 5 .Finally, conclusions and potential future work are given in Section 6.

2 RELATED WORKS

Each document in a corpus corresponds to an m-dimensional vector d, where m is the total number of terms that the document corpus 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.

The principle definition of clustering is to arrange data objects into separate clusters such that the intra-cluster similarity as well as the inter-cluster dissimilarity is maximized. The problem formulation itself implies that some forms of measurement are needed to determine such similarity or dissimilarity. There are many state-of-threat clustering approaches that do not employ any specific form of measurement, for instance, probabilistic model based method , non-negative matrix factorization , information theoretic co-clustering and so on. In this paper, though, we primarily focus on methods that indeed do utilize a specific measure. In the literature, Euclidean distance is one of the most popular measures:

$Dist\ (di, dj) = |\ |di\ -\ dj\ ||$

It is used in the traditional k-means algorithm. The objective of k-means is to minimize the Euclidean distance between objects of a cluster and that cluster's centroid:

$$\min \sum_{r=1}^{k} \sum_{di\ \in\ Sr} \|\ d_i - C_r\ \|^2$$

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 similarity score in text mining and information retrieval [12]. Particularly, 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:

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

Cosine measure is used in a variant of k-means called spherical k-means [3]. 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:

$$\mathbf{\max \sum_{r=1}^{k} \sum_{d_i \in Sr} \frac{d^t_i Cr}{\|\ Cr\|}}$$

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 directions. 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. The min-max cut graph-based spectral method is an example [13]. In graph partitioning approach, 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.

In nearest-neighbor graph clustering methods, such as the CLUTO's graph method above, the concept of similarity is somewhat different from the previously discussed methods. Two documents may have a certain value of cosine similarity, but if neither of them is in the other one's neighborhood, they have no connection between them. In such a case, some context-based knowledge or relativeness property is already taken into account when considering similarity. Recently, Ahmad and Dey [21] proposed a method to compute distance between two categorical values of an attribute based on their relationship with all other attributes. Subsequently, Ienco et al. [22] introduced a similar context-based distance learning method for categorical data. However, for a given attribute, they only selected a relevant subset of attributes from the whole attribute set to use as the context for calculating distance between its two values. More related to text data, there are phrase-based and concept-based document similarities. Lakkaraju et al. [23] employed a conceptual tree-similarity measure to identify similar documents. This method requires representing documents as concept trees with the help of a classifier. For clustering, Chim and Deng [24] proposed a phrase-based document similarity by combining suffix tree model and vector space model. They then used Hierarchical Agglomerative Clustering algorithm to perform the clustering task. However, a drawback of this approach is the high computational complexity due to the needs of building the suffix tree and calculating pairwise similarities explicitly before clustering. There are also measures designed specifically for capturing structural similarity among XML documents [25]. They are essentially different from the document-content measures that are discussed in this paper.

In general, cosine similarity still remains as the most popular measure because of its simple interpretation and easy computation, though its effectiveness is yet fairly limited. In the following sections, we propose a novel way to evaluate similarity between documents, and consequently formulate new criterion functions for document clustering.

3. SIMILARITY MEASURES

Before clustering, a similarity/distance measure must be determined.The measure reflects the degree of closeness or separation of the target objects and should correspond to the characteristics that are believed to distinguish the clusters embedded in the data. In many cases, these characteristics are dependent on the data or the problem context at hand, and there is no measure that is universally best for all kinds of clustering problems.

Moreover, choosing an appropriate similarity measure is also crucial for cluster analysis, especially for a particular type of clustering algorithms. For example, the density-based clustering algorithms, such as DBScan [4], rely heavily on the similarity computation. Density-based clustering finds clusters as dense areas in the data set, and the density of a given point is in turn estimated as the closeness of the corresponding data object to its neighboring objects. Recalling that closeness is quantified as the distance/similarity value, we can see that large number of distance/similarity computations are required for finding dense areas and estimate cluster assignment of new data objects. Therefore, understanding the effectiveness of different measures is of great importance in helping to choose the best one.

In general, similarity/distance measures map the distance or similarity between the symbolic description of two objects into a single numeric value, which depends on two factors— the properties of the two objects and the measure itself. In order to make the results of this study comparable to previous research, we include all the measures that were tested in [17] and add another one—the averaged Kullback-Leibler divergence. These five measures are discussed below. Different measure not only results in different final partitions, but also imposes different requirements for the same clustering algorithm, as we will see in Section 4.

3.1 Metric

Not every distance measure is a metric. To qualify as a metric, a measure d must satisfy the following four conditions.

Let x and y be any two objects in a set and d(x, y) be the distance between x and y.

1. The distance between any two points must be nonnegative,
that is, $d(x, y) \geq 0$.

2. The distance between two objects must be zero if and only if the two objects are identical, that is, $d(x, y) = 0$ if and only if $x = y$.

3. Distance must be symmetric, that is, distance from x to y is the same as the distance from y to x, ie.
$d(x, y) = d(y, x)$.

4. The measure must satisfy the triangle inequality, which
is $d(x, z) \leq d(x, y) + d(y, z)$.

3.2 Euclidean Distance

Euclidean distance is a standard metric for geometrical problems. It is the ordinary distance between two points and can be easily measured with a ruler in two- or three-dimensional space. Euclidean distance is widely used in clustering problems, including clustering text. It satisfies all the above four conditions and therefore is a true metric. It is also the default distance measure used with the K-means algorithm.

Measuring distance between text documents, given two documents da and db represented by their term vectors $\overrightarrow{ta}$ and $\overrightarrow{tb}$ respectively, the Euclidean distance of the two documents is defined as

$$DE(\overrightarrow{t_a}, \overrightarrow{t_b}) = \Big(\sum_{t=1}^{m} | w_{t,a} - w_{t,b}|^2\Big)^{1/2},$$

where the term set is T = {t1, . . . , tm}. As mentioned previously, we use the tfidf value as term weights, that is $w_{t,a}$ = tfidf(da, t).

3.3 Cosine Similarity

When documents are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors. This is quantified as the cosine of the angle between vectors, that is, the so-called cosine similarity. Cosine similarity is one of the most popular similarity measure applied to text documents, such as in numerous information retrieval applications [21] and clustering too [9].

Given two documents $\vec{ta}$ and $\vec{tb}$, their cosine similarity is

$$\mathbf{SIM_C}\,(\vec{\mathbf{ta}},\ \vec{\mathbf{tb}}) = \frac{\vec{\mathbf{ta}} \cdot \vec{\mathbf{tb}}}{|\,\vec{\mathbf{ta}}\,| \times |\,\vec{\mathbf{tb}}\,|}$$

Where $\vec{ta}$ and $\vec{tb}$ are m-dimensional vectors over the term set T = {t1, . . . , tm}. Each dimension represents a term with its weight in the document, which is non-negative. As a result, the cosine similarity is non-negative and bounded between [0,1].

An important property of the cosine similarity is its independence of document length. For example, combining two identical copies of a document d to get a new pseudo document d^1, the cosine similarity between d and d^1 is 1, which means that these two documents are regarded to be identical. Meanwhile, given another document l, d and d^1 will have the same similarity value to l,

that is, sim($\vec{t_d}$, $\vec{t_l}$) = sim($\vec{t_{d1}}$, $\vec{t_l}$). In other words, documents with the same composition but different totals will be treated identically. Strictly speaking, this does not satisfy the second condition of a metric, because after all the combination of two copies is a different object from the original document. However, in practice, when the term vectors are normalized to a unit length such as 1, and in this case the representation of d and d^1 is the same.

3.4 Jaccard Coefficient

The Jaccard coefficient, which is sometimes referred to as the Tanimoto coefficient, measures similarity as the intersection divided by the union of the objects. For text document, the Jaccard coefficient compares the sum weight of shared terms to the sum weight of terms that are present in either of the two document but are not the shared terms. The formal definition is:

$$\mathbf{SIM_J}\,(\vec{\mathbf{ta}},\ \vec{\mathbf{tb}}) = \frac{\vec{\mathbf{ta}} \cdot \vec{\mathbf{tb}}}{|\,\vec{\mathbf{ta}}\,|^2 \times |\,\vec{\mathbf{tb}}\,|^2 - \ \vec{\mathbf{ta}} \cdot \vec{\mathbf{tb}}}$$

The Jaccard coefficient is a similarity measure and ranges between 0 and 1. It is 1 when the $\vec{ta} = \vec{tb}$ and 0 when ta and tb are disjoint, where 1 means the two objects are the same and 0 means they are completely different. The corresponding distance measure is DJ = 1 − SIM_J and we will use D_J instead in subsequent experiments.

3.5 Averaged Kullback-Leibler Divergence

In information theory based clustering, a document is considered as a probability distribution of terms. The similarity of two documents is measured as the distance between the two corresponding probability distributions. The Kullback- Leibler divergence (KL divergence), also called the relative entropy, is a widely applied measure for evaluating the differences between two probability distributions.

Given two distributions P and Q, the KL divergence from distribution P to distribution Q is defined as

$$\mathbf{D_{KL}(P||Q) = Plog(\frac{P}{Q})}$$

In the document scenario, the divergence between two distribution of words is:

$$\mathbf{D_{KL}}(\vec{\mathbf{ta}}||\vec{\mathbf{tb}}) = \sum \mathbf{w_{t,a}} \times \mathbf{log}\left(\frac{\mathbf{w_{t,a}}}{\mathbf{w_{t,b}}}\right)$$

However, unlike the previous measures, the KL divergence is not symmetric, ie.
$D_{KL}(P||Q) \neq D_{KL}(Q||P)$. Therefore it is not a true metric. As a result, we use the averaged KL divergence instead, which is defined as

$$\mathbf{DAvgKL(P||Q) = \pi_1 DKL(P||M) + \pi_2 DKL(Q||M),}$$

$$\textbf{where}\quad \pi_1 = \frac{\mathbf{P}}{\mathbf{P+Q}},\quad \pi_2 = \frac{\mathbf{Q}}{\mathbf{P+Q}},\quad \textbf{and}\ \ \mathbf{M = \pi_1 P + \pi_2 Q.}$$

The average weighting between two vectors ensures symmetry, that is, the divergence from document i to document j is the same as the divergence from document j to document i. The averaged KL divergence has recently been applied to clustering text documents, such as in the family of the Information Bottleneck clustering algorithms [18], to good effect.

3.6 novel similarity measure

The cosine similarity can be expressed in the following form without changing its meaning:

$$\mathbf{Sim(di, dj) = cos(di{-}0, dj{-}0) = (di{-}0)^t\,(dj{-}0)}$$

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 is, if we look at them from many different viewpoints. From a third point d_h, the directions and distances to d_i and d_j are indicated respectively by the difference vectors $(d_i - d_h)$ and $(d_j - d_h)$. By standing at various reference points dh to view d_i, d_j and working on their difference vectors, we define similarity between
the two documents as:

$$\underset{d_i,d_j \in s_r}{\mathbf{Sim}(d_i,d_j)} = \frac{1}{n-n_r} \sum_{d_h \in s\setminus s_r} \mathbf{Sim}(d_i - d_h, d_j - d_h)$$

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 similarity 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 d_i and d_j by MVS(d_i, d_j | d_i, d_j Є Sr), or occasionally MVS(d_i, d_j) for short.

The final form of MVS in Eq. depends on particular formulation of the individual similarities within the sum. If the relative similarity is defined by dot-product of the difference vectors, we have:

$$\mathbf{MVS}(d_i, d_j \,|d_i, d_j \in Sr) = \frac{1}{n-n_r} \sum_{d_h \in S\setminus Sr} (d_i,d_h)^t(d_j-d_h)$$

$$= \frac{1}{n-n_r} \sum_{d_h} \cos(d_i-d_h,d_j-d_h)\|d_i - d_h\| \, \|d_j - d_h\|$$

The similarity between two points di and dj inside cluster Sr, viewed from a point dh outside this cluster, is equal to the product of the cosine of the angle between di and dj looking from dh and the Euclidean distances from dh 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. The smaller the distances ||di−dh|| and ||dj −dh|| are, the higher the chance that dh is in fact in the same cluster with d_i and d_j, and the similarity based on dh should also be small to reflect this potential. Therefore, through these distances, also provides a measure of intercluster dissimilarity, given that points d_i and d_j belong to cluster Sr, whereas dh belongs to another cluster. The overall similarity between d_i and d_j is determined by taking average over all the viewpoints not belonging to cluster Sr. It is possible to argue that while most of these viewpoints are useful, there may be some of them giving misleading information just like it may happen with the origin point. However, given a large enough number of viewpoints and their variety, it is reasonable to assume that the majority of them will be useful. 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.

3.7 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.

```
procedure BUILDMVSMATRIX(A)
  for r ← 1 : c do
      $D_{S\setminus Sr} \leftarrow \sum_{di \notin Sr} di$
      $n_{S\setminus Sr} \leftarrow |S \setminus Sr|$
  end for
  for i ← 1 :n do
      r ← class of $d_i$
      for j ← 1 : n do
          if dj Є Sr then
              $a_{ij} \leftarrow d^t_i d_j - d^t_i \frac{D_{S\setminus Sr}}{n_{S\setminus Sr}} - d^t_j \frac{D_{S\setminus Sr}}{n_{S\setminus Sr}} + 1$
          else
              $a_{ij} \leftarrow d^t_i d_j - d^t_i \frac{D_{S\setminus Sr}}{n_{S\setminus Sr}} - d^t_j \frac{D_{S\setminus Sr}}{n_{S\setminus Sr}} + 1$
          end if
      end for
  end for
  return $A=\{a_{ij}\}_{n\times n}$
end procedure
```

Fig. 1. Procedure: Build MVS similarity matrix.

From this condition, it is seen that even when dl is considered "closer" to d_i in terms of CS, i.e.
cos(d_i, d_j)≤cos(d_i, d_l), d_l can still possibly be regarded as less similar to d_i based on MVS if, on the contrary, it is "closer" enough to the outer centroid $C_{S\setminus Sr}$ than d_j is. This is intuitively reasonable, since the "closer" d_l is to $C_{S\setminus Sr}$, the greater the chance it actually belongs to another cluster rather than Sr and is, therefore, less similar to di. 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 documents that are closest to it based on this measure should be in the same cluster with it.The validity test is designed as following. For each type of similarity measure, a similarity matrix A $=\{a_{ij}\}n\times n$ is created. For CS, this is simple, as $a_{ij} = d^t_i d_j$ .The procedure for building MVS matrix is described in Fig. 1. Firstly, the outer composite w.r.t. each class is determined. Then, for each row a_i of A, i = 1, . . . , n, if the pair of documents d_i and d_j, j = 1, . . . , n are in the same class, a_{ij} is calculated as in line 10, Fig. 1. Otherwise, dj is assumed to be in d_i's class, and a_{ij} is calculated as in line 12, Fig. 1. After matrix A is formed, the procedure in Fig. 2 is used to get its validity score. For each document d_i corresponding to row a_i of A, we select q_r documents closest to d_i. The value of q_r is chosen relatively as percentage of the size of the class r that contains di, where percentage ∈ (0, 1]. Then, validity w.r.t. di is calculated by the fraction of these q_r documents having the same class label with di, as in line 12, Fig. 2. The final validity is determined by averaging

```
Require: $0 < percentage \le 1$
1: procedure GETVALIDITY(validity,A, percentage)
2:  for $r \leftarrow 1 : c$ do
3:      $qr \leftarrow [percentage \times nr]$
4:      if $qr = 0$ then
5:          $qr \leftarrow 1$
6:      end if
7: end for
8:  for $i \leftarrow 1 : n$ do
9:       $\{aiv[1], . . . , aiv[n]\} \leftarrow$ Sort $\{ai1, . . . , ain\}$
10:       s.t. $aiv[1] \ge aiv[2] \ge . . . \ge aiv[n]$
          $\{v[1], . . . , v[n]\} \leftarrow$ permute $\{1, . . . , n\}$
11:   $r \leftarrow$ class of $di$
12:  $validity(di) \leftarrow \frac{|\{dv[1], . . . , dv[qr]\} \cap Sr|}{qr}$
13: end for
14: $validity \leftarrow \frac{\sum^{n}_{i\leftarrow 1} validity(di)}{n}$
15:  return validity
16:  end procedure
```

Fig. 2. Procedure: Get validity score

over all the rows of A, as in line 14, Fig. 2. It is clear that validity score is bounded within 0 and 1. The higher validity score a similarity measure has, the more suitable it should be for the clustering task.

Two real-world document datasets are used as examples in this validity test. The first is reuters7, a subset of the famous collection, Reuters-21578 Distribution 1.0, of Reuter's newswire articles1. Reuters-21578 is one of the most widely used test collection for text categorization. In our validity test, we selected 2,500 documents from the largest 7 categories: "acq", "crude", "interest", "earn", "money-fx", "ship" and "trade" to form reuters7. Some of the documents may appear in more than one category. The second dataset is k1b, a collection of 2,340 web pages from the Yahoo! subject hierarchy, including 6 topics: "health", "entertainment", "sport", "politics", "tech" and "business". It was created from a past study in information retrieval called WebAce [26], and is now available with the CLUTO toolkit [19].

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.

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 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.

4.MULTI-VIEWPOINT BASED CLUSTERING

Having defined our similarity measure, we now formulate 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 F:

$$F = \sum^{k} n_r \left[1 / n^2_r \sum_{d_i,d_j \text{€} Sr} Sim(d_i,d_j) \right]$$

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 and effective way. Let us use a parameter α called the regulating factor, which has some constant value (α Є [0, 1]), and let $\lambda r = n^{\alpha}_r$ in Eq. the final form of our criterion function I_R is:

$$I_R = \sum_{r=1}^{k} \frac{1}{n_r^{1-\acute{\alpha}}} \left[\frac{n+n_r}{n-n_r} \|Dr\|^2 - \left(\frac{n+n_r}{n-n_r} - 1\right) D^t_r D\right]$$

In the empirical study of Section 5.4, it appears that I_R's performance dependency on the value of α is not very critical. The criterion function yields relatively good clustering results for α Є (0, 1).

In the formulation of IR, a cluster quality is measured by the average pairwise similarity between documents within that cluster. However, such an approach can lead to 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 dense cluster. Though not as clear as with CS, it is still possible that the same effect may hinder MVS-based clustering if using pairwise similarity. To prevent this, an alternative approach is to consider similarity between each document vector and its cluster's centroid instead.

4.1 Optimization algorithm and complexity

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

$$I_V = \sum_{r=1}^{k} Ir\ (nr,Dr)$$

where Ir (nr,Dr) 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 documents are selected to be the seeds from which initial 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 kmeans 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 nz 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 nz 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).

5 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.

5.1 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 thorough 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 stopword removal, stemming, removal of too rare as well as too frequent words, TF-IDF weighting and normalization.

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

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

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

Our MVSC-I_R and MVSC-I_V programs are implemented in Java. The regulating factor α in I_R is always set at 0.3 during the experiments. We observed that this is one of the most appropriate values. A study on MVSC-I_R'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 initializationdependent. Hence, for each method, we performed clustering
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 external 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 "recall"(R) values used in information retrieval. Given a clustering solution, FScore is determined as:

$$\mathbf{FScore=\sum_{i=1}^{k}\frac{n_i}{n_j}\max(F_{i,j})}$$

where ni denotes the number of documents in class i, n_j the number of documents assigned to cluster j, and $n_{i,j}$ 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.

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

$$\mathbf{Accuracy=\frac{1}{n_q}\max\sum_{i=1}^{k}n_{i,q(i)}}$$

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.

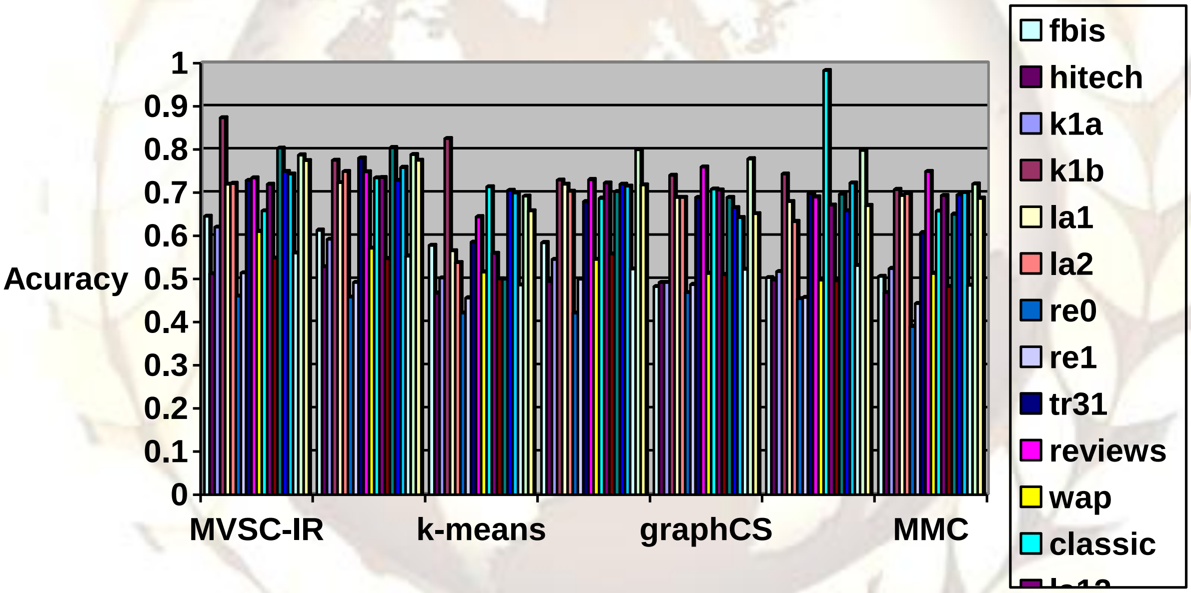
5.3 Results

Fig. 6 shows the Accuracy of the seven clustering algorithms 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 perform 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 performer is Spkmeans. The other algorithms might work well on certain dataset. For example, graphEJ yields
outstanding result on classic; graphCS and MMC are 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-IR and

MVSC-IV 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-IR, k-means), from Table 3, it is seen that MVSC-IR 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 MVSCIR and MVSC-IV over the other methods is statistically significant. A special case is the graphEJ algorithm. On the one hand, MVSC-IR is not significantly better than graphEJ if based on FScore or NMI. On the other hand,

when MVSC-IR and MVSC-IV are tested obviously better 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 considered as outliers, and including them in the statistical tests would affect the outcomes greatly. Hence, w e also report in Table 5 the tests where classic was excluded and only results on the other 19 datasets were used.



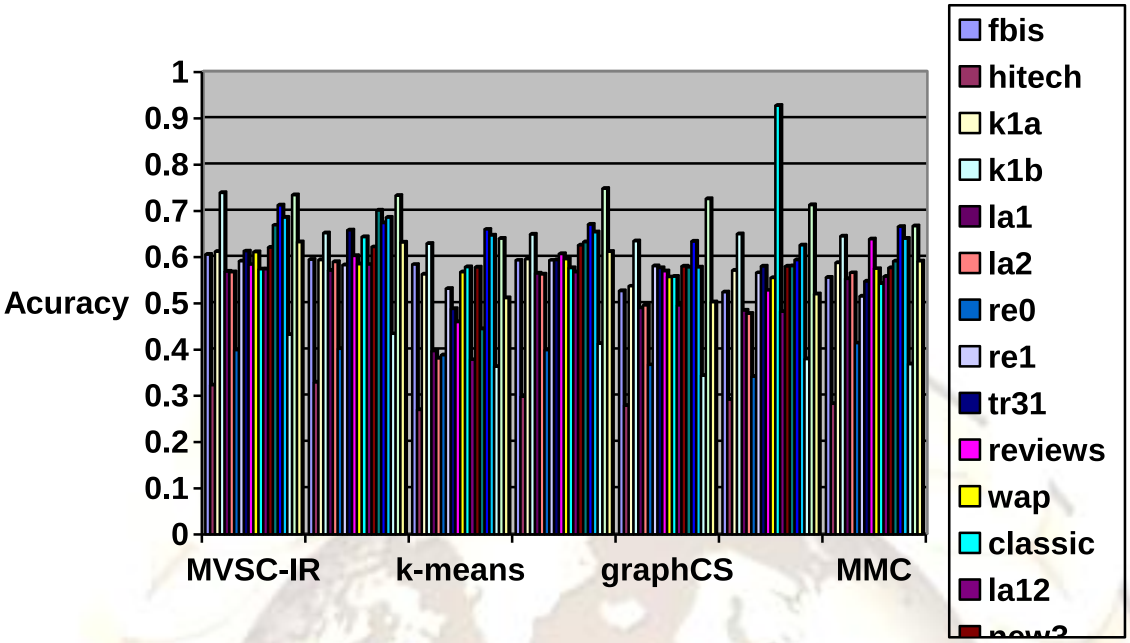


Fig. 6. Clustering results in Accuracy. Left-to-right in legend corresponds to left-to-right in the plot.

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	.503	.506
hitech	.512	.528	.467	.494	.492	.497	.468
k1a	.620	.592	.502	.545	.492	.517	.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	.510	.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	.595	.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 outperform graphEJ significantly with good p-values.

5.4 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 I_R , there exists parameter α which is called the regulating factor, α Є [0, 1]. To examine how the determination of α could affect MVSC-I_R'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 measure of MVSC-IR with $\alpha = \alpha_i$ is determined as following

$$\textit{relative FScore}\ (IR; S, \alpha i) = \frac{\max_{\alpha j}\{FScore(IR; S, \alpha j)\}}{FScore(IR; S, \alpha i)}$$

where αi, αj ∈ {0.0, 0.1, . . . , 1.0}, FScore(I_R; S, αi) is the FScore result on dataset S obtained by MVSC-I_R with α = αi. 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.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we analyses 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. Based on MVS, two criterion functions, I_R and I_V , and their respective clustering algorithms, MVSC-I_R and MVSC-I_V , have been introduced. 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.

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 or the relative similarity , or do not use average but have other methods to combine the relative similarities according to the different 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 hierarchical 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.

REFERENCES

[1] D. Arthur and S. Vassilvitskii. k-means++ the advantages of careful seeding. In Symposium on Discrete Algorithms, 2007.

[2] M. Craven, D. DiPasquo, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. In AAAI-98, 1998.

[3] D. R. Cutting, J. O. Pedersen, D. R. Karger, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In Proceedings of the ACM SIGIR, 1992.

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of 2nd International Conference on KDD, 1996.

[5] N. Friburger and D. Maurel. Textual similarity based on proper names. In Proceedings of Workshop on Mathematical Formal Methods in Information Retrieval at th 25th ACM SIGIR Conference, 2002.

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

[8] D. Lee and 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, and 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, and Y. Gong, "Document clustering based on nonnegative matrix factorization," in SIGIR, 2003, pp. 267–273.

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

[12] C. D. Manning, P. Raghavan, and H. Sch ¨ utze, An Introduction to Information Retrieval. Press, Cambridge U., 2009.

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

[14] H. Zha, X. He, C. H. Q. Ding, M. Gu, and H. D. Simon, "Spectral relaxation for k-means clustering," in NIPS, 2001, pp. 1057–1064.

[15] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, pp. 888–905, 2000.

[16] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in KDD, 2001, pp. 269–274.

[17] Y. Gong and W. Xu, Machine Learning for Multimedia Content Analysis. Springer-Verlag New York, Inc., 2007.

[18] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," Mach. Learn., vol. 55, no. 3, pp. 311–331, Jun 2004.

[19] G. Karypis, "CLUTO a clustering toolkit," Dept. of Computer Science, Uni. of Minnesota, Tech. Rep., 2003, http://glaros.dtc.umn.edu/gkhome/views/cluto.

[20] A. Strehl, J. Ghosh, and R. Mooney, "Impact of similarity measures on web-page clustering," in Proc. of the 17th National Conf. on Artif. Intell.: Workshop of Artif. Intell. for Web Search. AAAI, Jul. 2000, pp. 58–64.

[21] A. Ahmad and L. Dey, "A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set," Pattern Recognit. Lett., vol. 28, no. 1, pp. 110 – 118, 2007.

[22] D. Ienco, R. G. Pensa, and R. Meo, "Context-based distance learning for categorical data clustering," in Proc. of the 8th Int. Symp. IDA, 2009, pp. 83–94.

[23] P. Lakkaraju, S. Gauch, and M. Speretta, "Document similarity based on concept tree distance," in Proc. of the 19th ACM conf. on Hypertext and hypermedia, 2008, pp. 127–132.

[24] H. Chim and X. Deng, "Efficient phrase-based document similarity for clustering," IEEE Trans. on Knowl. and Data Eng., vol. 20, no. 9, pp. 1217–1229, 2008.

[25] S. Flesca, G. Manco, E. Masciari, L. Pontieri, and A. Pugliese, "Fast detection of xml structural similarity," IEEE Trans. on Knowl. And Data Eng., vol. 17, no. 2, pp. 160–175, 2005.

[26] E.-H. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore, "Webace: a web agent for document categorization and exploration," in AGENTS '98: Proc. of the 2nd ICAA, 1998, pp. 408–415.