International Journal of Machine Learning and Cybernetics (2022) 13:663–675
https://doi.org/10.1007/s13042-021-01295-8

ORIGINAL ARTICLE

Multi-view document clustering based on geometrical similarity measurement

Bassoma Diallo[1] · Jie Hu[1] · Tianrui Li[1] · Ghufran Ahmad Khan[1] · Ahmed Saad Hussein[1,2]

Received: 15 July 2020 / Accepted: 8 March 2021 / Published online: 22 March 2021

Abstract
Numerous works implemented multi-view clustering algorithms in document clustering. A challenging problem in document clustering is the similarity metric. Existing multi-view document clustering methods broadly utilized two measurements: the Cosine similarity (CS) and the Euclidean distance (ED). The first did not consider the magnitude difference (MD) between the two vectors. The second can't register the divergence of two vectors that offer a similar ED. In this paper, we originally created five models of similarity metric. This methodology foils the downside of the CS and ED similarity metrics by figuring the divergence between documents with the same ED while thinking about their sizes. Furthermore, we proposed our multi-view document clustering plan which dependent on the proposed similarity metric. Firstly, CS, ED, triangle's area similarity and sector's area similarity metric, and our five similarity metrics have been applied to every view of a dataset to generate a corresponding similarity matrix. Afterward, we ran clustering algorithms on these similarity matrices to evaluate the performance of single view. Later, we aggregated these similarity matrices to obtain a unified similarity matrix and apply spectral clustering algorithm on it to generate the final clusters. The experimental results show that the proposed similarity functions can gauge the similitude between documents more accurately than the existing metrics, and the proposed clustering scheme surpasses considerably up-to-date algorithms.

Keywords Multi-view clustering · Ensemble clustering · Similarity measurement · Document clustering

1 Introduction

Documents clustering (DC) is an instinctive management of learning task, which groups high correlation documents into same category and divides those of disparate into different categories simultaneously [1]. Recently DC [2–4, 6–9, 12–16] turns into an intriguing issue. The typical structure of DC comprises of text refining and knowledge distillation. During the former step, a procedure changing a document into an intermediate form can be document-based or concept-based [17]. In the next stage, clustering algorithms are then applied to extract valuable information according to the intermediate form. To take out a decent example from the document, scientists utilized diverse machine learning algorithms. In past years, various works applied text mining techniques to analyze the text patterns and carry out their mining process [19]. Researchers categorized these techniques into agglomerative clustering algorithms, partitioning algorithms, and standard parametric modeling-based methods [20]. Automatic document organization, topic extraction, fast information retrieval or filtering are the common document clustering applications [21].

One key issue in document clustering is the similitude estimation [22–25]. There are two widely used geometrical similarity metrics: the Cosine Similarity (CS) [26–28] and the Euclidean distance (ED) [29]. The former suffers from the magnitude difference (MD) of vectors which is

✉ Jie Hu
jiehu@swjtu.edu.cn

Bassoma Diallo
sanediallo2003@yahoo.fr

Tianrui Li
trli@swjtu.edu.cn

Ghufran Ahmad Khan
ghufraan.alig@gmail.com

Ahmed Saad Hussein
husseinsaad187@yahoo.com

[1] Institute of Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China

[2] University of Information Technology and Communications, Baghdad 00964, Iraq

dealing with term frequency. CS computes the similarity level between two documents without taking into count the rated frequency of each term. The subsequent one neglects to calculate the difference of two vectors that offer a similar ED and does not perform well on high-dimensional data. In this manner, these two measurements are not consistently appropriate to figure out the similarity between several documents.

Another issue in document clustering is how to group documents derived from different sources. Due to the diversity and massive volume of unorganized text documents generated from diverse sources [30], extracting high quality of information from text documents is a challenging task. A dataset is referred to as multi-view data when it derives from diverse sources or pattern [31]. Data from different sources have dissimilar physical connotations and statistical properties. To describe the divergent information, several studies regarded each source or modality as one "view". In that way, multi-view learning can be utilized to join information which will produce better results in one hand. In another way, it can be used to minimize the effect of noise (data values that make it harder to find patterns) in the data. The properties as mentioned above make the multi-view learning a great candidate to be used in text document clustering. In the last decade, several works were concentrated on the multi-view document clustering challenges [2–4, 6, 26, 32]. Some of them consider all the data points in an individual view [3]; others regard them as multiple views [32]. Multi-view Clustering (MvC) aims to accurately and robustly partition the data more than any single-view clustering [33]. Thus, two methods are available, firstly, the distributed method where views are clustered separately, then fused the results to get a final partition. Secondly, simultaneously fuses all views into one, then apply clustering algorithms. The method called centralization suffers from the over-fitting problem and ignores the statistical property of each view. Merging multiple views without decreasing the accuracy is an ongoing challenge in multi-view clustering. Recently [35] reported that there is no criterion to decide which MvC algorithm is the best. To that end, ensemble clustering can be applied to MvC to take advantage of different methods.

In this article, we extend our past research work [36]. Knowing that existing geometrical approaches of similarity measurement consider the magnitude, the ED, the cosine and the direction of vectors separately and inspired by the previous research results, the current work proposes a Robust Multi-view Document Clustering (RMDC) method to address the similarity measurement, the fusion of documents derived from multiple sources challenges in text document clustering. The major differences between the Concept-Enhanced Multi-view Clustering (CEMvC) [36], and RMDC are summarized as follows. Based on the theory that every metric has its advantage and disadvantage according to the dataset, we explore a new theory which not only extended CEMvC but improved it consequently. We instantiate five models of metric instead of three in the previous work. To that end, we run them on every dataset to determine which metric is more suitable. We apply text preprocessing mentioned in Sect. 4.2 on all texts in each dataset and create a list of top n keywords with high TF-IDF score from each benchmark dataset. The algorithm applies the CS, ED, TS-SS, and RDSim1-5 similarity metrics to the data matrix of each view to generate the corresponding similarity matrices. Furthermore, an ensemble approach is deployed to combine these matrices into a solid similarity matrix for the final clustering process. Then a partition-based algorithm such as spectral clustering is deployed to cluster the data. RMDC considers more views in the dataset whereas our previous work focused only on two views per dataset. More rigorous analysis on every metric as well as dataset are conducted respectively in Sects. 6.3 and 6.5, which show the robustness of the proposed RMDC. The main contributions of our work are as following:

- The proposed RMDC tackles the drawbacks of CS and ED metrics by calculating the similarity between documents with the same ED while taking into consideration their MD.
- Our method does not only compute the similarity between documents but also their similarity level.

The rest of this paper is structured as follows: the Sect. 2 surveys the related works on text document clustering and the ensemble clustering method. In Sect. 3, we review some basic notions of similarity metric and multi-view methods. We present our proposed multi-view document clustering scheme in Sect. 4. In Sect. 5 we analyze the time complexity of the proposed algorithm. We conduct experimental studies in Sect. 6. Section 7 contains the present paper conclusion and plan of the future work.

2 Related works

This section presents a review of recent literature on document clustering as well as multi-view document clustering which are the foundation of our proposed method.

Recently, extensive studies on document clustering have been carried out. Priya and Priyadharshini [7] proposed a new algorithm named text clustering with feature selection. This algorithm identifies pertinent features (i.e., terms) by iteratively incorporating an improved supervised feature

selection to identify important features. In their proposed framework, they represented the terms such as synonym, meronym or hypernym and concept relationship in the ontology. The proposed algorithm in [13] works well in small data, but need to be upgraded to deal with large-scale document datasets. Yu et al. [37] combined text mining techniques and the bibliometric methods to analyze the patterns of the information science publications, geographic distribution, source journals, source institutes, international collaboration, inter-institutional collaboration, document co-citation network, and the references citation bursts detection. Saini et al. [38] fused the self-organizing map (SOM) and the multi-objective differential evolution approach yielding to a cognitive-inspired multi-objective automatic document clustering technique. They utilized the concept of SOM to design new genetic operators for the proposed clustering technique. Furthermore, they encoded the variable number of cluster centers in different solutions of the population to automatically determine the number of clusters from a data set. Sherkat et al. [11] proposed an innovative visual analytic scheme for reciprocal document clustering. In the proposed system, introductory clustering is established based on the user-defined number of clusters and the preferred clustering algorithm. A set of coordinated visualizations allow the examination of the dataset and the results of the clustering. The visualization provides the user with the highlights of individual documents and understanding of the evolution of documents over the time period to which they relate. The users then interact with the process by means of changing key-terms that drive the process according to their knowledge of the document's domain. In key-term-based synergy, the user designates a set of keywords to each object cluster to instruct the clustering algorithm. We have improved that process with a novel algorithm for choosing proper seeds for the clustering. Janani and Vijayarani [5] proposed an improved text document clustering framework based on the combination of Spectral Clustering algorithm with Particle Swarm Optimization (SCPSO). By the use of global and local optimization function the algorithm aims to deal with the huge volume of text documents. However, the complexity of the similarity graph matrix generation is very high. The method in [6] barely deals with the overlapping clusters and the matrix generation problems. Abualigah et al. [15] combined several objective functions and algorithms such as Krill Herd. They initially inherit solutions from the k-mean clustering algorithm and the clustering agreement, then combined the two objective functions. Bisson and Grimal [2] proposed the Multi-view similarity (MVSIM) framework to handle the dilemma of learning co-similarities when a set of matrices describes the connection between various items. To handle noise in the data, they set the percentage parameter p of the smallest similarity values to be zero in the document and word matrices at the end of each iteration. One drawback in their method is that this parameter relies on prior information which is not accurate. The second drawback is that it processes noise during the clustering step which might affect the performance.

Multi-view document clustering emerged to address the problem of grouping documents derived from diverse sources. [35, 45, 46] discussed the recent progress and new challenges regarding multi-view clustering. Zhao et al. [46] categorized multi-view learning mechanisms into three majors: co-training style algorithms, co-regularization style algorithms, and margin-consistency style algorithms. Lastly, Yang and Wang [35] classified the learning method into multi-kernel learning, multi-view subspace, multi-task multi-view, multi-view graph clustering, and co-training technique algorithms. According to their study, the correctness of views, the opportune moment of fusion, the incomplete MvC, and the multi-task multi-view clustering are still challenging problems. Furthermore, [47–50] advised some new trend directions. Wahid et al. [6] proposed a non-dominated sorting genetic multi-view document clustering based algorithm. This method generates distinctive clustering solutions from the multiple views of the documents and then privileges a mixture of clusters to form a final clustering. Hussain et al. [3] combined different ensemble techniques which lead to a novel multi-view document clustering algorithm. Their algorithm computes three particular similarity matrices on each dataset and aggregates them to set up a consensual similarity matrix, which is then used as an input of a clustering algorithm to obtain the final clustering. However, their algorithm is computationally expensive, and its accuracy relies on the multiple clustering algorithms used. Inspired by this work, in this study, we extend the same idea then compute the similarity matrices in a parallel to drop down the computation cost. We detailed our framework in Sect. 4. Furthermore, the same authors proposed a multi-view clustering setting in the context of a co-clustering framework [32] based on the assumption that transferring similarity values from one view to the others regarding the individual data will enhance a clustering result. They extended a co-clustering algorithm named χ-SIM[1] to multi-view clustering. However, this method suffers from the problem of executing the number of iterations accurately. The multiple views detection in documents is still a challenging

[1] https://sites.google.com/site/fawadsyed/

Table 1 List of symbols

Symbol	Description
u	Document 1
v	Document 2
d	Dimensionality space
D	The total number of document
$cos(u,v)$	The cosine between document u and v
$ED(u,v)$	The Euclidean distance between document u and v
$TS(u,v)$	The triangle's area similarity between document u and v
$SS(u,v)$	The sector's area similarity between document u and v
$MD(u,v)$	The magnitude difference between document u and v
n	The number of views
m	The number of metrics
M^i	Aggregated similarity matrix of view i
M^i_j	Similarity matrix of view i computed by metric j
M^f_{sim}	The unified similarity matrix
C	The number of documents clusters

problem [51]. The multi-view concept factorization (MVCF) [8] technique incorporates a graph-regularized method to cluster document. The MVCF algorithm preserves the local geometrical structure of the manifolds for multi-view clustering what the traditional concept factorization can not. The proposed algorithm is only suitable for small scale datasets and has its time complexity is very high. To overcome this problem, Jia et al. [10] devised an approximate normalized cuts algorithm beyond the eigen-decomposition for large scale clustering. Firstly, they reduced the space prerequisite of the normalized cut by sampling a few data points to deduce the global features of dataset instead of using the full affinity matrix. Secondly, they accelerated the graph cut clustering proceeding in an iterative way that using the approximate weighted kernel k-means to optimize the objective function of normalized cut. This technique avoids the direct eigen-decomposition of Laplacian matrix.

Similarly, Yan et al. [9] proposed a novel regularized concept factorization algorithm, which focuses on two constraints. Firstly, whether two documents belong to the same class (must-connected) and secondly when they are in different classes (cannot-connected). It is well-known that there is no criterion to decide which MvC algorithm is the best. One way to take advantage of them is to combine them in an ensemble learning method what we discuss below.

3 Preliminaries

This section briefly reviewed three commonly used similarity metrics in document clustering analysis.

Notations: Let V be the data matrix representing a dataset having documents in rows and words in its columns. V^i_j denotes the elements of V that corresponds to the intensity of association between document i and word j. For simplicity, we select two documents u and v in the data view. Table 1 gives a detailed summary of the notation used through this paper.

3.1 Similarity metric

The measure of similarity between two documents is a complex task in text mining. Several studies proposed some similarity measurements. Birjali et al. [22] suggested a Map Reduce-based algorithm to measure the similarity in a large corpus document. In their study, they discussed similarity measures based on the arcs, nodes, vector space, and hybrids. Wagh and Anand [23] compared two approaches for finding legal document similarity: (a) CS, (b) citation based similarity. The most difference is the use of Jaccard similarity in the citation-based similarity case. Results show that citation-based similarity measure is more robust in determining parallel among cases but requires more connected components. Jagatheeshkumar and Brunda [24] surveyed about similarity measures based on distance such as ED, Manhattan distance, Minkowski distance, CS. Shirkhorshidi et al. [25] examined the role of them on high-dimensional datasets.

We introduce the two most commonly used similarity distances which are CS and ED.

3.1.1 Cosine similarity

CS formulated in Eq. (1) computes the pairwise similarity between two documents using dot product and magnitude of vector document **u** and vector document **v** in high-dimensional space [52–56].

$$Cos(u,v) = \frac{\sum_{n=1}^{k} u(n).v(n)}{|u|.|v|} \tag{1}$$

CS refers to as a metric for measuring distance when the MD of the vectors is not prerequisite. Text data represented by word counts is a suitable case to apply this technique. For example, a group of word occurs more in one text document as it is longer than the other text document which is shorter in length. In this case, the weight of this community might be more substantial for the first document than the second, but they appear to be similar documents. In such cases, CS would be a better metric.

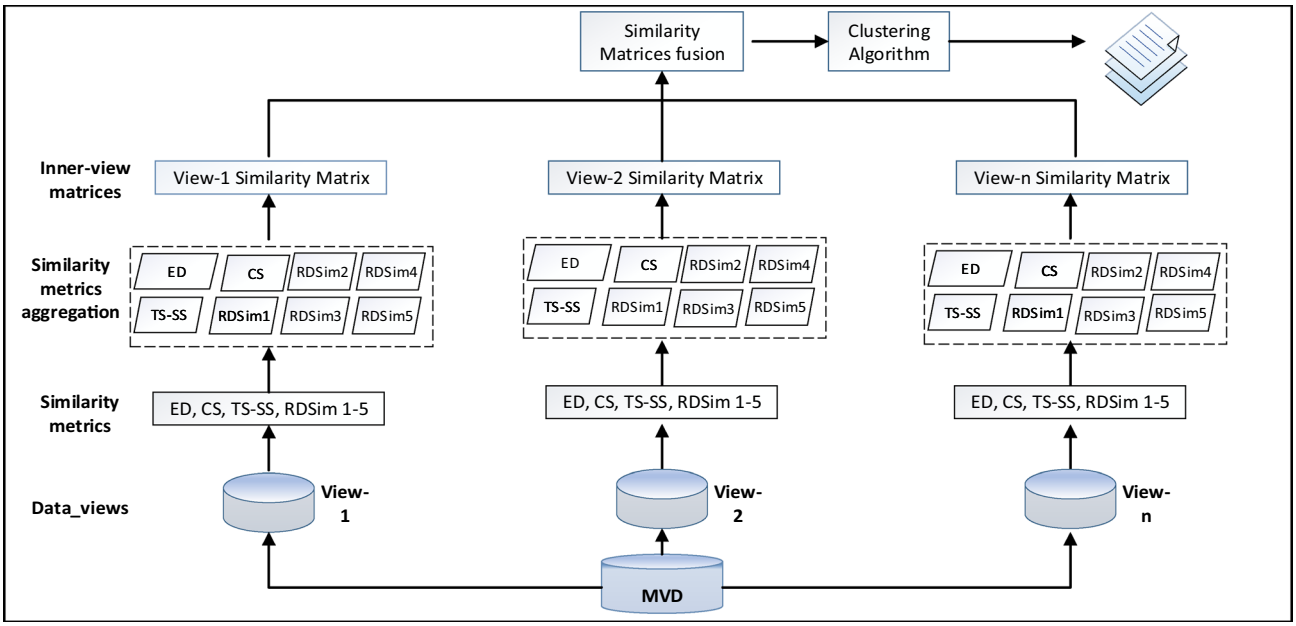


Fig. 1 Framework for the proposed robust multi-view document clustering

3.1.2 Euclidean distance

The ED [57, 58] within two points a and b is the portion of the straight-line distance connecting them. In this case, it appears to be the second extensively used similarity metric.

$$ED(u,v) = \sqrt{\sum_{n=1}^{k} (u(n) - v(n))^2} \tag{2}$$

where u is the first document and v the second one.

ED computes in n-dimensional, the distance between two points space based on their coordinate.

3.1.3 TS-SS similarity

Heidarian and Dinneen highlighted both drawbacks of CS and ED in [14], and then they proposed a new method named TS-SS[2]. The method combines the triangle's area similarity TS formulated as follows:

$$TS(u,v) = \frac{|u|.|v|.\sin\left(\theta'\right)}{2} \tag{3}$$

and the sector's area similarity SS depicted in the next formula.

$$SS(u,v) = \frac{\pi.\left(\theta'\right)[ED(u,v) + MD(u,v)]^2}{360} \tag{4}$$

TS-SS is formulated as follows:

$$TS - SS(u,v) = \frac{\pi.|u|.|v|.\theta'.\sin(\theta').(ED(u,v) + MD(u,v))^2}{720} \tag{5}$$

where $MD(u,v) = \left|\sqrt{\sum_{n=1}^{k} u_n^2} - \sqrt{\sum_{n=1}^{k} v_n^2}\right|$ and $\theta' = \cos^{-1}(\cos(u,v)) + 10$

4 Proposed method

In this work, we first instantiate five models of metric named Robust Document Similarity metric ($RDSim_{1-5}$). Then we advise our multi-view document clustering scheme based on the proposed new similarity metrics.

4.1 Document similarity metrics

CS and ED metric are not always suitable to measure the similarity between two documents. CS is known to be one of the good geometric similarity measurements. However, it does not consider the MD of the two vectors. Both CS and

[2] https://github.com/taki0112/

ED are limited to estimate the similarity between two documents accurately. Knowing that they complete each other, an alternative is to combine them. Therefore, there is a need to build a novel approach to calculate similarity which can cope with the drawbacks of these metrics. To that end, we devise the following robust document similarity metrics ($RDSim_{1-5}$):

$$RDSim1(u,v) = ED(u,v) \times Cos(u,v) + TS - SS(u,v) \quad (6)$$

$$RDSim2(u,v) = [ED(u,v) + MD(u,v)]Cos(u,v) \quad (7)$$

$$RDSim3(u,v) = [SS(u,v) + Cos(u,v)]ED(u,v) \quad (8)$$

$$RDSim4(u,v) = [SS(u,v) + Cos(u,v)]TS(u,v) \quad (9)$$

$$RDSim5(u,v) = ED(u,v) \times SS(u,v) \quad (10)$$

The RDSim1 metric in Eq. (6) computes the similarity between two documents by taking into consideration their ED, Cosine, the triangle's as well as the sector's area similarities. Since the ED is sometimes large, for the RDSim2 metric in Eq. (7), we strengthen the cosine with the sector's and the triangle's area similarity. From RDSim 3–5, we revise the ED metric with the sector's area similarity in Eq. (8), the MD and the cosine in Eq. (9) and combined the sector's area similarity with the cosine in Eq. (10).

After computing the document similarity matrix with the above five metrics, we devise a method to aggregate different matrices generated from several views in each document.

4.2 Multi-view document cluster ensemble

We propose an ensemble technique to combine different similarity matrix generated from the metrics as mentioned above in the following steps:

- Step 1: Document preprocessing
  In document preprocessing, Tokenization is a crucial step and refers to partition the document into an array of sentences which in turn into words. Following the processing in [59], we use the word_tokenize function of the natural language toolkit (nltk) to tokenize the words. This procedure generates many words which affect the clustering accuracy. We consider words like stop-words which are not precise enough as noise, and establishes a collection of irrelevant similarities. These words have to be pruned or removed accordingly. Porter's stemming algorithm is then applied to reduce inflected words into their stems. Later, TF-IDF is applied to measure the term frequency, then filter the words that appear with very low frequency throughout the corpus. To compute the similarity between two documents A and B, we convert their sentences into vectors with TF-IDF. The vectors are then equalized to the same length. These data serve as the input for the afterward step.
- Step 2: Similarity matrices generation
  On every dataset, the equalized list of view is used as the input to *CS* using Eq. (1), *ED* Eq.(2), TS-SS Eq.(5), and $RDSim_{1-5}$ similarity metrics to generate the corresponding view similarity matrix. For instance, in View1, the output is dataset Name-View1-Cosine-sim-matrix, dataset Name-View1-ED-sim-matrix, dataset Name-View1-TS-SS-Cosine-sim-matrix etc. For n views, we obtain $8n$ matrices.
- Step 3: Inner-view similarity matrices aggregation
  We concatenate the matrix generated by every metric view by view to obtain the inner-view matrix. We repeat the procedure above to all the views. Then, we combine in the next step the individual inner-view matrix for each view to improve the clustering using the formula in Eq. (11)

$$M^v = \frac{1}{n}\sum_{i=1}^{n}(M_j^i + M_r^i) \quad (11)$$

  where for view i, M_j^i is the matrix generated by the traditional geometrical similarity metrics j, M_r^i the matrix produced by our five proposed metrics r and n is the total number of document views.
- Step 4: Inter-view similarity matrices aggregation
  We aggregate the inner-view similarity matrices to obtain a unified final similarity matrix. For $n > 2$ views, our proposed algorithm aggregates n ensemble based similarity matrices. In this paper we fix the number n to 3.
- Step 5: Final clustering
  The final similarity matrix which is then used as the input of clustering algorithms such as spectral clustering to generate the final clusters C. The clustering performance is evaluated using accuracy 12 and purity 13 evaluation metrics. The overall procedure is highlighted in Fig. 1, and the pseudo-code is displayed in Algorithm 1.

Algorithm 1: Robust Multi-view Document Clustering (RMDC)

Input: Set of data view: $\{X_1; X_2; ...X_n\}$,
number of cluster K,
number of document, number of metrics m,
i is a set of *views*,
j is a set of *metrics* $\{Cosine, ED, TS-SS\}$,
r is a set of *metrics* $(RDSim_{1-5})$

Output: The final similarity matrix: M^f_{sim}; the final cluster C

***Step1**: Calculate the similarity matrices M^i_j*
for *i in views* **do**
  while *j in metrics {Cosine, ED, TS − SS}* **do**
    Compute corresponding similarity matrices M^i_j using Eq.(1, 2, 5)
    Return matrix M^i_j
  end
  while *r in metrics {RDSim$_{1-5}$}* **do**
    Compute corresponding similarity matrices M^i_r using Eq.(6, 7, 8, 9, 10)
    Return matrix M^i_r
  end
end
***Step2**: Inner-View Similarity Matrices Aggregation*
foreach *view v* **do**
  Aggregate the similarity matrices employing Eq.(11)
  Return matrix M^v
end
***Step3**: Cross-View Similarity Matrices Aggregation*
for *v in view* **do**
  Aggregate the inter-view similarity matrices to obtain the unified matrix $M^f_{sim} = \frac{1}{n}\sum M^v$
  Return M^f_{sim}
  Run spectral clustering algorithm on M^f_{sim} to obtain the final clusters C
  Return C
end

5 Time complexity analysis

The complexity of the proposed method relies on the RDSim algorithm used during the similarity matrices generation. Given a dataset with n views $n \geq 2$, the objects number q and m similarity measurements, the overall complexity is $O(qnmd)$ where d is the data dimensionality. To save the memory we compute the similarity matrix in parallel and store in the disk. This approach makes easy to reuse the same matrix without recomputing it again.

Table 2 A description of datasets

Data sets	Objects	Features	Clusters	Views
Citeseer	3312	8435 (3703 + 2366 + 2366)	6	3
Cora	2708	6862 (1433 + 2714 + 2715)	7	3
Cornell	195	2272 (1703 + 284 + 285)	5	3
Texas	187	2281 (1703 + 289 + 289)	5	3
Washington	230	2486 (1703 + 392 + 391)	5	3
Wisconsin	265	2686 (1703 + 469 + 469)	5	3

6 Experiments

In this section, we conduct tests on six (6) real-world multi-view datasets to evaluate the effectiveness of the proposed approach. We run all the experiments in PYTHON3 on a work-station (Windows 64bits, Intel(R) Core (TM) i7-4600 CPU @2.10 GHz 2.70 GHz processors, 16GB of RAM).

6.1 Data sets description

We present each dataset by specifying the views and features in Table 2 . In all cases the content view is the documents-words matrix, containing 0/1 values indicating absence or presence of a word in a document. The inbound view is the matrix indicating by 0/1 values describing the inbound links between documents. The cites view is the matrix of the number of citation links between documents.

CiteSeer[3] The dataset contains 3312 documents over the 6 labels (Agents, IR, DB, AI, HCI, ML). Every document is made of the following views: content, inbound, cites. The documents are described by 3703 words in the content view, and by the 4732 links between them in the inbound, and cites views.

Cora[4] contains 2708 documents over the 7 labels (Neural Networks, Rule Learning, Reinforcement Learning, Probabilistic Methods, Theory, Genetic Algorithms, Case Based). It is made of same number of views like Citesser dataset. It is described by the absence/presence of the word in a set of publication as the first view in the dataset. The second view consists of citation links to scientific publications. The documents are described by 1433 words in the content view, and by the 5429 links between them in the inbound, and cites views.

Cornell[5] contains 195 documents over the 5 labels (student, project, course, staff, faculty). It is made of 3 views (content, inbound, cites) on the same documents. The documents are described by 1703 words in the content view, and

[3] https://linqs-data.soe.ucsc.edu/public/lbc/citeseer.tgz
[4] http://www.cs.umd.edu/~sen/lbc-proj/data/cora.tgz
[5] http://membres-lig.imag.fr/grimal/data/Cornell.tar.gz

by the 569 links between them in the inbound, and cites views.

Texas[6] is one of the four subsets of WEBKB dataset. The first view is a matrix of document-by-words while the second view corresponds to document-links and is one of the four universities datasets. The documents are described by 1703 words in the content view, and by the 578 links between them in the inbound, and cites views. The dataset documents belongs to 5 different classes(student, project, course, staff, faculty).

Washington[7] contains 230 documents over the five labels (student, project, course, staff, faculty). It is made of the views of content, inbound, and cites on every documents. A set of 1703 words describe the documents in the first view, and 783 links between them in the other views.

Wisconsin[8] is an archive of 265 documents over the five labels (student, project, course, staff, faculty). It is made of 3 views (content, inbound, and cites) on the same documents. The documents consist of 1703 words in the content view, and 938 links between them in the inbound, and cites views.

6.2 Evaluation metric

Knowing that an external index measure the agreement between two partitions where the first partition is the priori known clustering label, and the second results from the predicting clustering procedure [18]. We employ the most two widely used external validity indices: Accuracy and Purity to evaluate clustering performance.

- Accuracy [34] measures how the set of predicted labels for a sample must exactly match the corresponding set of true labels. Accuracy is defined as follows:

$$ACC = \frac{1}{n}\sum_{i=1}^{n}\delta\left(map\left(c_i\right), g_i\right) \tag{12}$$

where n is the total number of samples, g_i is the ground-truth label, $\delta(u, v)$ is the delta function that equals 1 for similar documents and equals 0 for the dissimilar one, $map(c_i)$ is the permutation mapping function that maps each cluster label c_i to the equivalent label from the data set.

- Purity [16] is an external evaluation criterion of cluster quality. It quantifies the extent that cluster C_i contains points only from one (ground truth) partition in the unit range from 0 to 1. The expression of the purity can write as follows:

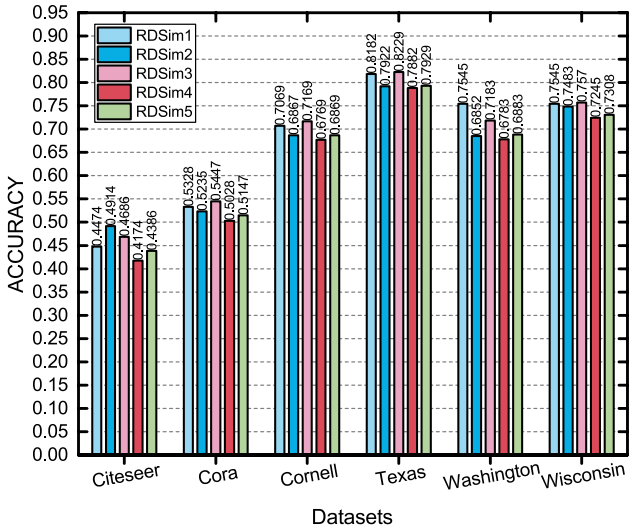


Fig. 2 Performance of RDSim1-5 metrics

$$Purity = \frac{1}{D}\sum_{i=1}^{k}\max_{j=1}^{k}\left\{p_{ij}\right\} \tag{13}$$

where D is the number of all documents in the dataset, k is the number of clusters, p_{ij} is the probability that a member of cluster j belongs to class i.

6.3 Analysis of the proposed similarity metrics

In all the experiment, we use spectral clustering, and the number of clusters k is equal to the real cluster number of the original dataset. Firstly, we evaluate the performance of the traditional geometrical similarity metrics.

Secondly, we compare the accuracy values among the different variations of RDSim. The accuracy results values of these metrics are shown in Table 3.

From Fig. 2 it can be observed that every metric has its advantage and disadvantage according to the dataset.

Table 3 reveals that $RDSim_3$ and $RDSim_5$ excel on Washington dataset. Citeseer seems to be the most challenging dataset for our metric. One can see that $RDSim_2$ metric performs better on Citeseer, Cornell, Texas datasets. We deduce that $RDSim_3$ and $RDSim_5$ are better on data where the variety of documents/texts is more important, and $RDSim_2$ is better when this variety is lower.

The overall evaluation for the 8 metrics is shown in Fig. 3. Among the proposed metrics $RDSim_5$ yields a poor accuracy in all dataset comparing to the others. This is due to the fact that we do not take into account the cosine while computing the similarity. So, it corroborates the hypothesis that cosine is important but not enough to measure the similarity between documents. $RDSim_4$ surpasses $RDSim_5$

[6] http://membres-lig.imag.fr/grimal/data/Texas.tar.gz

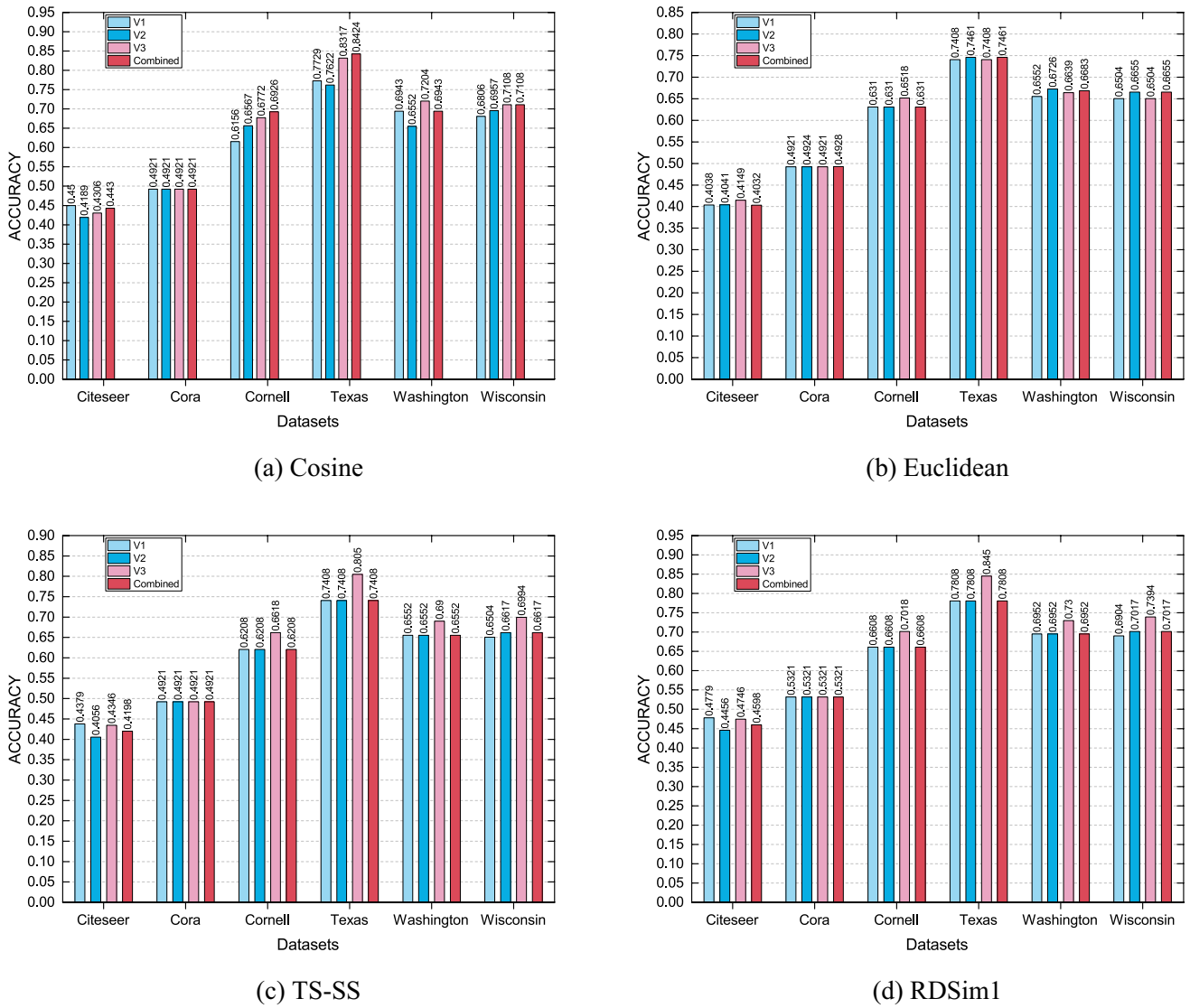
[7] http://lig-membres.imag.fr/grimal/data/Washington.tar.gz

[8] http://lig-membres.imag.fr/grimal/data/Wisconsin.tar.gz

Table 3 Evaluation of the 8 similarity metrics on the multi-view datasets in term of accuracy

Datasets	Cosine similarity	Euclidean distance	TS-SS	RDSim1	RDSim2	RDSim3	RDSim4	RDSim5
Citeseer	0.4337	0.4032	0.4198	0.4598	**0.4737**	0.4561	0.4598	0.4561
Cora	0.4921	0.4928	0.4921	**0.5321**	**0.5321**	**0.5321**	**0.5321**	**0.5321**
Cornell	0.6208	0.6310	0.6208	**0.6608**	**0.6608**	0.6556	**0.6608**	0.6556
Texas	0.7836	0.7461	0.7408	0.7808	**0.8129**	0.7808	0.7808	0.7808
Washington	0.6552	0.6683	0.6552	0.6952	0.7126	**0.7170**	0.6952	**0.7170**
Wisconsin	0.7183	0.6655	0.6617	0.7017	**0.7319**	0.6942	0.7017	0.6942

The best values are highlighted in bold



(a) Cosine

(b) Euclidean

(c) TS-SS

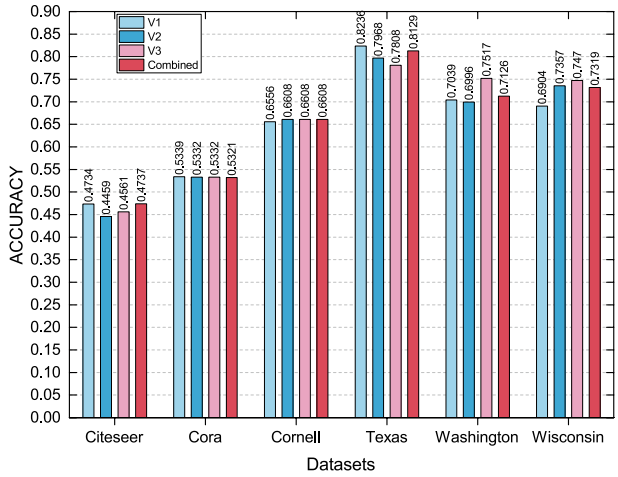
(d) RDSim1

Fig. 3 Comparison of the 8 similarity metrics on 6 multi-view datasets

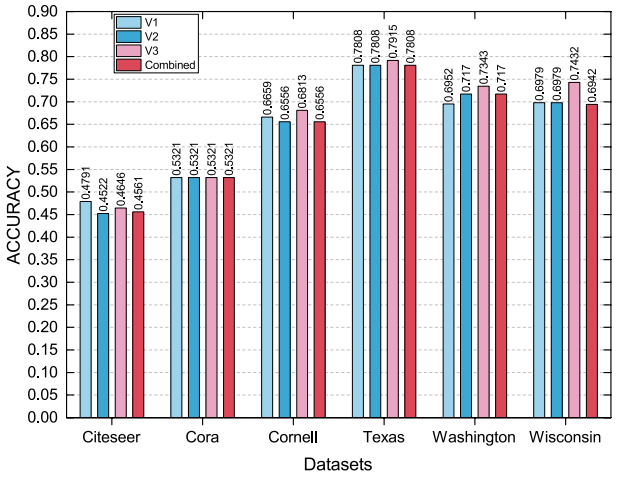
while still can not excel the other metric since it ignores the ED during the similarity computation. This result confirms the hypothesis based on ED. To that end, it appear that combining ED and CS and boost the document similarity measurement.

6.4 RMDC comparison with other algorithms

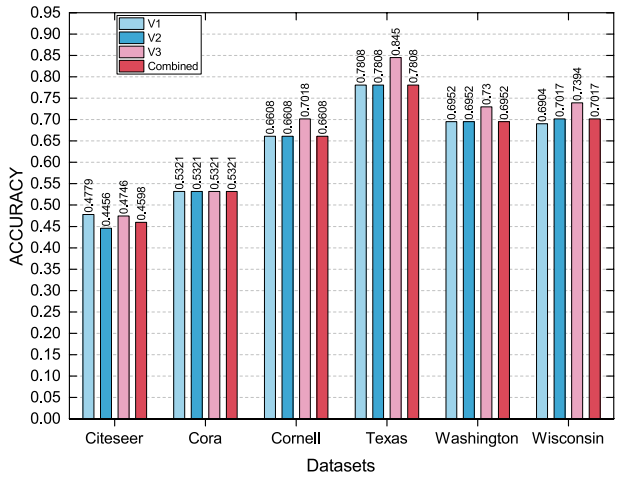
We compare our proposed method to the following algorithms:



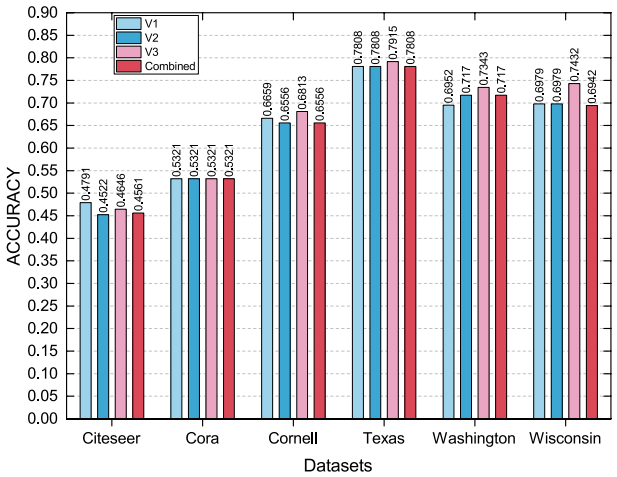
(e) RDSim2



(f) RDSim3



(g) RDSim4



(h) RDSim5

Fig. 3 (continued)

- Multi-view ensemble clustering (MVEC) [3]: The algorithm computes three different similarity matrices named cluster-based similarity matrix, affinity matrix and pairwise dissimilarity matrix on the individual datasets and aggregates these matrices to form a combined similarity matrix, which serves as the input of a final clustering algorithm.
- Multi-view concept factorization (MVCF) [8]: the algorithm identifies the underlying coefficient matrices for each view, and then fuses them with a multi-manifold regularizer to locally conserve the data geometrical format while learning the individual view weights automatically.
- NMF model with co-orthogonal constraints (NMF-CC) [33] : NMF-CC adds a co-orthogonal constraint to the representation and basis matrices for further capturing the diversity within each view and learning the appropriate basis matrices, in which the basis vector is independent to each other.
- Cluster-based similarity partitioning algorithm (CSPA) [60]: The algorithm detects the relation among objects in the equivalent cluster by inducing a similarity measure from the partitioning. Further, calculates the pairwise similarity between them and then reclusters each object by using this similarity measurement to determine the combined clustering.
- Weighted hybrid clustering (WHC) [61]: the algorithm first computes a weighted kernel fusion clustering based on voting techniques to calculate individual clustering results from each data, then combines them using a weighted ensemble clustering technique;

Table 4 Accuracy evaluation of the proposed method compared to the state-of-the-art document clustering algorithms

Method	Citeseer	Cora	Cornell	Texas	Washington	Wisconsin
RMDC	0.4598	0.5121	**0.6808**	**0.7918**	**0.6802**	**0.7660**
NMF-CC [33]	0.4539	0.5001	0.6379	0.6936	0.6797	0.74.09
KMLRSSC [66]	0.4432	0.4951	0.5246	0.5043	0.5330	0.5928
DiNMF [65]	–	–	0.5446	0.5777	0.5930	0.6203
MMNMF [67]	0.4489	0.4815	0.4308	0.5775	0.4739	0.6147
SCaMVC [68]	–	–	0.4256	0.5508	0.4652	–
CaKMVC [69]	–	–	0.4615	0.5775	0.6000	–
SAMVC-H [70]	–	–	0.4927	0.6115	0.5761	–
SAMVC-S [70]	–	–	0.4979	0.6343	0.6115	–
MVCF [8]	–	**0.5576**	0.6769	–	–	–
kCC [64]	–	0.4800	0.6300	–	0.6600	–
MVEC [3]	**0.4964**	0.4431	0.4715	0.5241	–	–
HEC [3]	0.4321	0.3394	0.4564	0.4278	–	–
WHC [3]	0.2630	0.2555	0.3744	0.3797	–	–
CSPA [3]	0.1932	0.1706	0.2513	0.3476	–	–
HCC [3]	0.2129	0.3010	0.4410	0.4064	–	–

The best values are highlighted in bold

- Hierarchical ensemble clustering (HEC) [62]: The objective of this algorithm is to connect partition-based and hierarchical clustering. The algorithm uses a set of dendrogram and aggregates them into a distance matrix. A consensus distance is then used to build a structured hierarchy on top of the consensus clustering.
- Hierarchical combination clustering (HCC) [63]: The algorithm consists of combining results from multi-views using hierarchical clustering. To that end, it converted this hierarchical clustering into matrices which describe the dendrogram distances and then aggregated them into a final matrix and used it for the combined clustering.
- *k*-means based co-clustering (kCC) [64]: The algorithm uses a greedy approach but only guarantees the local optimum solution.
- Diverse NMF (DiNMF) [65]: DiNMF utilizes a diversity term to explore diversity of from different views. This approach has two parameters, which are selected identical with the original literature.
- Kernel Multi-view low-rank sparse subspace clustering (KMLRSSC) [66] : KMLRSSC[9] is a spectral based multi-view clustering method with low-rank and sparsity constraints, where the centroid-based scheme is used to learn the consensus matrix.
- Multi-view clustering via multi-manifold regularized non-negative matrix factorization (MMNMF) [67]: MMNMF incorporates consensus manifold and consensus coefficient matrix with multi-manifold regularization to preserve the locally geometrical structure of the multi-view data space.
- Multi-view clustering with soft capped norm (SCaMVC) [68]: SCaMVC learns an optimal weight for each view automatically without introducing an additive parameter as previous methods do. Furthermore, to deal with different level noises and outliers, it uses soft-capped norm, which caps the residual of outliers as a constant value and provides a probability for certain data point being an outlier.
- Multi-view capped-norm *k*-means clustering (CaKMVC) [69]: CaKMVC utilizes the capped-norm based residual calculation for the objective to remove the effects of the outliers.
- Self-paced and auto-weighted multi-view clustering (SAMVC) [70]: SAMVC learns the MVC model with easy examples and then progressively considers complex ones from each view. In addition, a soft weighting scheme of self-paced learning is designed to further reduce the negative impact from outliers and noises.

We quote the results from the original paper [32, 70] for algorithms whose codes are not available publicly. Compared to previous works, our method outperforms the other models as shown in Table 4 on four of the six multi-view document data.

6.5 Analysis of the proposed multi-view document clustering method

We analyze the accuracy scores of the proposed multi-view document clustering algorithm on each benchmark dataset.

[9] https://github.com/Geovhbn/MLRSSC

Comparing to the state-of-the-art algorithm, Table 4 shows that RMDC outperforms with significant marge. It can be observed that our proposed RMDC performs better on Cornell, Texas, Washington and Wisconsin datasets than the previous methods. The Citeseer dataset shows the least accuracy among all the algorithms tested. We speculate that this might be due to the fact that a variety of document in this dataset is higher than the others and this data is more subject to noise.

7 Conclusion and future work

In this work, a robust multi-view document Clustering have been proposed. To that end, we instantiated five similarity measurements and concatenate these similarity metrics to solve the problem of similarity measurement in document clustering. Our metrics calculate the dissimilarity between documents based on their Cosine, Euclidean distances and MD. The similarity matrices are computed in parallel to diminish the computation cost. Furthermore, we recommended a robust multi-view clustering method tailored to cluster documents. The experimental analysis shows that every metric of $RDSim_{1-5}$ has its advantage and disadvantage according to the dataset and the RMDC approach exceeds diverse advanced multi-view clustering schemes. Despite its good results, the proposed method consumes more space and ran slower when the data dimensionality increases. In our future work, we will overcome this issue by combining diverse dimensionality reduction approaches.

Acknowledgements This work is supported by the National Science Foundation of China (nos. 61772435, 61976182, 61876157) and the Fundamental Research Funds for the Central Universities (no. 220710004005040177) and Sichuan Key R&D project (no. 2020YFG0035).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Shah N, Mahajan S (2012) Document clustering: a detailed review. Int J Appl Inf Syst 4(5):30–38
2. Bisson G, Grimal C (2012) Co-clustering of multi-view datasets: a parallelizable approach. In: Proceedings of the 12th international conference on data mining. IEEE, pp 828–833
3. Hussain SF, Mushtaq M, Halim Z (2014) Multi-view document clustering via ensemble method. J Intell Inf Syst 43(1):81–99
4. Sabthami J, Thirumoorthy K, Muneeswaran K (2016) Multi-view clustering of clinical documents based on conditions and medical responses of patients. In: Proceedings of the 10th international conference on intelligent systems and control (ISCO). IEEE, pp 1–5
5. Janani R, Vijayarani S (2019) Text document clustering using spectral clustering algorithm with particle swarm optimization. Proc Expert Syst Appl 134:192–200
6. Wahid A, Gao X, Andreae P (2014) Multi-view clustering of web documents using multi-objective genetic algorithm. In: Proceedings of the congress on evolutionary computation (CEC). IEEE, pp 2625–2632
7. Priya MJS (2012) Clustering technique in data mining for text documents. Int J Comput Sci Inf Technol 1:2943–2947
8. Zhan K, Shi J, Wang J, Tian F (2017) Graph-regularized concept factorization for multi-view document clustering. J Vis Commun Image Represent 48:411–418
9. Yan W, Zhang B, Ma S, Yang Z (2017) A novel regularized concept factorization for document clustering. Knowl Based Syst 135:147–158
10. Jia H, Ding S, Du M, Xue Y (2016) Approximate normalized cuts without Eigen-decomposition. Inf Sci 374:135–150
11. Sherkat E, Milios EE, Minghim R (2019) A visual analytic approach for interactive document clustering. ACM Trans Interact Intell Syst 10(1):1–33
12. Hussain SF, Bisson G, Grimal C (2010) An improved co-similarity measure for document clustering. In: Proceedings of the 9th international conference on machine learning and applications, 2010, pp 190–197
13. Xu S, Chan K-S, Gao J, Xu X, Li X, Hua X, An J (2016) An integrated k-means-Laplacian cluster ensemble approach for document datasets. Neurocomputing 214:495–507
14. Heidarian A, Dinneen MJ (2016) A hybrid geometric approach for measuring similarity level among documents and document clustering. In: Proceedings of the 2nd international conference on big data computing service and applications. IEEE, pp 142–151
15. Abualigah LM, Khader AT, Hanandeh ES (2018) A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis. Eng Appl Artif Intell 73:111–125
16. Huang S, Xu Z, Lv J (2018) Adaptive local structure learning for document co-clustering. Knowl Based Syst 148:74–84
17. Tan AH, Ridge K, Labs D, Terrace HMK (1999) Text mining: the state of the art and the challenges," Proceedings of the Pakdd workshop on knowledge discovery from advanced databases, pp 65–70
18. Kaijun W, Baijie W, Liuqing P (2009) CVAP: Validation for cluster analyses. Data Sci J 0904220071–0904220071
19. Talib R, Kashif M, Ayesha S, Fatima F (2016) Text mining: techniques, applications and issues. Int J Adv Comput Sci Appl 7(11):414–418
20. Bhardwaj B (2016) Text mining, its utilities, challenges and clustering techniques. Int J Comput Appl 135(7):22–24
21. Yue L, Zuo W, Peng T, Wang Y, Han X (2015) A fuzzy document clustering approach based on domain-specified ontology. Data Knowl Eng 100:148–166
22. Birjali M, Beni-Hssane A, Erritali M (2016) Measuring documents similarity in large corpus using mapreduce algorithm. In: Proceedings of the 5th international conference on multimedia computing and systems. IEEE, 2016, pp 24–28
23. Wagh R, Anand D (2017) Application of citation network analysis for improved similarity index estimation of legal case documents: a study. In: International conference on current trends in advanced computing, (ICCTAC). IEEE, 2017, pp 1–5
24. Jagatheeshkumar G, Brunda SS (2017) An analysis of efficient clustering methods for estimates similarity measures. In: Proceedings of the 4th international conference on advanced computing and communication systems. IEEE, 2017, pp 1–3
25. Shirkhorshidi AS, Aghabozorgi S, Wah TY (2015) A comparison study on similarity and dissimilarity measures in clustering continuous data. PloS One 10(12):1–20

26. Popat SK, Deshmukh PB, Metre VA (2017) Hierarchical document clustering based on cosine similarity measure. In: Proceedings of the 1st international conference on intelligent systems and information management. IEEE, 2017, pp 153–159
27. George KK, Kumar CS, Sivadas S, Ramachandran K, Panda A (2018) Analysis of cosine distance features for speaker verification. Pattern Recognit Lett 112:285–289
28. Kalhori H, Alamdari MM, Ye L (2018) Automated algorithm for impact force identification using cosine similarity searching. Measurement 122:648–657
29. Diego JSN, Mesquita PP, João PP Gomes, Amauri HSJ (2017) Euclidean distance estimation in incomplete datasets. Neurocomputing 248:11–18
30. Sailaja NV, Padmasree L, Mangathayaru N (2016) Survey of text mining techniques, challenges and their applications. Int J Comput Appl 146(11):30–35
31. Ye Y, Liu X, Liu Q, Yin J (2017) Consensus kernel k-means clustering for incomplete multi-view data. Comput Intell Neurosci 2017:1–11
32. Hussain SF, Bashir S (2016) Co-clustering of multi-view datasets. Knowl Inf Syst 47(3):545–570
33. Liang N, Yang Z, Li Z, Sun W, Xie S (2020) Multi-view clustering by non-negative matrix factorization with co-orthogonal constraints. Knowl Based Syst 105582
34. Jin H, Feiping N, Heng H, Chris D (2014) Robust manifold nonnegative matrix factorization. ACM Trans Knowl Discov Data 8(3):1–21
35. Yang Y, Wang H (2018) Multi-view clustering: a survey. Big Data Min Anal 1(2):83–107
36. Diallo B, Hu J, Li T, Khan G, Ji C (2019) Concept-enhanced multi-view clustering of document data. In: Proceedings of the 14th international conference on intelligent systems and knowledge engineering. IEEE, 2019, pp 1357–1363
37. Yu D, Xu Z, Pedrycz W, Wang W (2017) Information sciences 1968–2016: a retrospective analysis with text mining and bibliometric. Inf Sci 418:619–634
38. Saini N, Saha S, Bhattacharyya P (2019) Automatic scientific document clustering using self-organized multi-objective differential evolution. Cogn Comput 11(2):271–293
39. Vega-Pons S, Ruiz-Shulcloper J (2011) A survey of clustering ensemble algorithms. Int J Pattern Recognit Artif Intell 25(03):337–372
40. Krawczyk B, Minku LL, Gama J, Stefanowski J, Woźniak M (2017) Ensemble learning for data stream analysis: a survey. Inf Fusion 37:132–156
41. Boongoen T, Iam-On N (2018) Cluster ensembles: a survey of approaches with recent extensions and applications. Comput Sci Rev 28:1–25
42. Xie X, Sun S (2013) Multi-view clustering ensembles. In: Proceedings of the 2013 international conference on machine learning and cybernetics. IEEE, 2013, pp 51–56
43. Cano A (2017) An ensemble approach to multi-view multi-instance learning. Knowl Based Syst 136:46–57
44. Huang S, Wang H, Li D, Yang Y, Li T (2015) Spectral co-clustering ensemble. Knowl Based Syst 84:46–55
45. Sun S (2013) A survey of multi-view machine learning. Neural Comput Appl 23(7–8):2031–2038
46. Zhao J, Xie X, Xu X, Sun S (2017) Multi-view learning overview: recent progress and new challenges. Inf Fusion 38:43–54
47. Jiang B, Qiu F, Wang L (2016) Multi-view clustering via simultaneous weighting on views and features. Appl Soft Comput J 47:304–315
48. Xu YM, Wang CD, Lai JH (2016) Weighted multi-view clustering with feature selection. Pattern Recognit 53:25–35
49. Huang S, Kang Z, Xu Z (2018) Self-weighted multi-view clustering with soft capped norm. Knowl Based Syst 158:1–8
50. Huang S, Kang Z, Tsang IW, Xu Z (2019) Auto-weighted multi-view clustering via kernelized graph learning. Pattern Recognit 88:174–184
51. Wahid A, Gao X, Andreae P (2015) Multi-objective clustering ensemble for high-dimensional data based on strength pareto evolutionary algorithm (spea-ii). In: Proceedings of the international conference on data science and advanced analytics. IEEE, 2015, pp 1–9
52. Xia P, Zhang L, Li F (2015) Learning similarity with cosine similarity ensemble. Inf Sci 307:39–52
53. Dong J-Y, Chen Y, Wan S-P (2018) A cosine similarity based qualiflex approach with hesitant fuzzy linguistic term sets for financial performance evaluation. Appl Soft Comput 69:316–329
54. Geng Z, Li Y, Han Y, Zhu Q (2018) A novel self-organizing cosine similarity learning network: an application to production prediction of petrochemical systems. Energy 142:400–410
55. Xiang W-L, Li Y-Z, He R-C, Gao M-X, An M-Q (2018) A novel artificial bee colony algorithm based on the cosine similarity. Comput Ind Eng 115:54–68
56. Moujahid D, Elharrouss O, Tairi H (2018) Visual object tracking via the local soft cosine similarity. Pattern Recognit Lett 110:79–85
57. Alencar J, Lavor C, Liberti L (2019) Realizing euclidean distance matrices by sphere intersection. Discrete Appl Math 256:5–10
58. Bapat RB, Kurata H (2019) On Cartesian product of Euclidean distance matrices. Linear Algebra Appl 562:135–153
59. Abasi AK, Khader AT, Al-Betar MA, Naim S, Makhadmeh SN, Alyasseri ZAA (2020) Link-based multi-verse optimizer for text documents clustering. Appl Soft Comput 87:Article 106002
60. Strehl A, Ghosh J (2002) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617
61. Liu X, Yu S, Moreau Y, Moor BD, Glänzel W, Janssens FAL (2009) Hybrid clustering of text mining and bibliometrics applied to journal sets. In: Proceedings of the international conference on data mining, 2009, pp 49–60
62. Zheng L, Li T, Ding C (2010) Hierarchical ensemble clustering. In: 10th international conference on data mining. IEEE, 2010, pp 1199–1204
63. Mirzaei H (2010) A novel multi-view agglomerative clustering algorithm based on ensemble of partitions on different views. In: Proceedings of the 20th international conference on pattern recognition. IEEE, 2010, pp 1007–1010
64. Hussain SF, Haris M (2019) A k-means based co-clustering (kCC) algorithm for sparse, high dimensional data. Expert Syst Appl 118:20–34
65. Wang J, Tian F, Yu H, Liu CH, Zhan K, Wang X (2018) Diverse non-negative matrix factorization for multi-view data representation. IEEE Trans Cybern 48(9):2620–2632
66. Brbić M, Kopriva I (2018) Multi-view low-rank sparse subspace clustering. Pattern Recognit 73:247–258
67. Zong L, Zhang X, Zhao L, Yu H, Zhao Q (2017) Multi-view clustering via multi-manifold regularized non-negative matrix factorization. Neural Netw 88:74–89
68. Huang S, Kang Z, Xu Z (2018) Self-weighted multi-view clustering with soft capped norm. Knowl Based Syst 158:1–8
69. Huang S, Ren Y, Xu Z (2018) Robust multi-view data clustering with multi-view capped-norm k-means. Neurocomputing 311:197–208
70. Ren Y, Huang S, Zhao P, Han M, Xu Z (2020) Self-paced and auto-weighted multi-view clustering. Neurocomputing 383:248–256