2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS) | 978-0-7381-2460-5/20/$31.00 ©2020 IEEE | DOI: 10.1109/IOTSMS52051.2020.9340213

A Comparative Study of Clustering Techniques Applied on Covid-19 Scientific Literature

Valerio Bellandi
Department of Computer Science,
Università degli studi di Milano,
Milan, Italy
Email: valerio.bellandi@unimi.it

Paolo Ceravolo
Department of Computer Science,
Università degli studi di Milano,
Milan, Italy
Email: paolo.ceravolo@unimi.it

Samira Maghool
Department of Computer Science,
Università degli studi di Milano,
Milan, Italy
Email: samira.maghool@unimi.it

Stefano Siccardi
Department of Computer Science,
Università degli studi di Milano,
Milan, Italy
Email: stefano.siccardi@unimi.it

***Abstract*—Due to the current emergency situation, caused by COVID-19, the scientific literature on the topic has rapidly grown. At the same time, purposeful and targeted research plans with strong background knowledge is urgently needed. However, the huge number of documents produced by multiple communities generates a fragmented terminology that may cause confusion in information retrieval. To this aim, in a comparative study, we test different techniques to efficiently cluster these publications for improving their level of findability.**

***Index Terms*—Text Embedding, Document Clustering, COVID_19, Machine Learning.**

I. Introduction

Document clustering, a field at the intersection between Natural Language Processing (NLP) and Machine Learning (ML), is widely used in organizing textual documents, to unveil unrecognized relationships among datasets or documents [1]. In recent months, the scientific community witnessed great effort in multidisciplinary studies regarding the COVID-19 pandemic, generating a huge and fragmented production. Due to the emergency situation, the urgent need is felt to pursue fast and purposeful researches. Insightful research needs strong background knowledge and ability in interconnecting the results achieved from ongoing projects. The published materials and their different relations, such as citations, common fields, and authors, could be assumed respectively as constituents nodes and links of a high-dimensional complicated graph. Machine Learning (ML) techniques can address the challenge of categorizing and organizing this massive data to get explainable and usable results. To this aim, text embeddings methods could play the most promising role. Generally, text embeddings algorithms represent words and documents as d-dimensional vectors focusing on preserving a similar context in the text body, efficiently close in the embedded space. The assigned vectors are defined by analyzing the body of text and converting each word, phrase, or the entire document's relations according to the similarity function applied in the embedding method. Therefore, a similar context in the body of texts located closely in the embedded space could be extracted using clustering.

In ML, Clustering is an unsupervised learning task highly used for exploratory data analysis to reveal some hidden patterns which are present in data but cannot be categorized visually. The idea is that data can be grouped based on some common characteristics. The mechanisms rely on the primary task of keeping instances with a large value of similarity, measured by some distance metric, in a cluster closer than instances belonging to other clusters.

The large variety of approaches and algorithms in clustering indicates the purpose dependency of clustering problems. Therefore, choosing appropriate clustering techniques and algorithms is determined by an understanding of the structure of the data, the kind of analysis to be carried out, and the size of the data set [2]. It was recognised that choosing appropriate clustering methods and the optimal number of clusters in healthcare data due to enormous amounts of data produced by electronic medical records, administrative reports, and other research findings [3] can be confusing and difficult.

Determining the quality of the results obtained by clustering techniques is a key issue in unsupervised machine learning. Some suggested indexes are measuring the quality of produced clusters based on compactness, separation, and distances from other clusters. Some of them also suggest combined formalism that at the same time, considers more than one of the mentioned factors.

In the presented paper, we aim to study different feature extracting techniques and different clustering algorithms for clustering some of COVID-19 publications for further use in information extraction. The K-means [4], DBSCAN [5], Agglomerative [6], MiniBatchkmeans [7] and BIRCH [8] algorithms due to their prominence in the field are chosen. Kmeans, Agglomerative, MiniBatchkmeans, and BIRCH require prior specification of the number of clusters while DBSCAN does not. First, in this work, we analyze K-means algorithm results using a different number of clusters (K). Secondly, we study DBSCAN algorithm using different the minimum number of points required to form a cluster ($Min\_samples$) and the Eps parameter for the radius of clusters. We assess the obtained results by three indexes, Silhouette [9], Davies_ Bouldin [10], [11] and calinski_harabasz [12] which drive the comparative analysis of the clustering algorithms we tested.

II. Related Works

A. Text Embedding

Information retrieval provides techniques to identify relevant information from a data collection [13]. Text Embedding in general relies on generating vector representations of documents, as proposed by Salton in 1971 [14]. In its simplest form, each document is represented by the (TF) vector, $vtf = (tf_1, tf_2, \ldots, tf_n)$, where tf_i is the frequency of the ith term in the document. Normally very common words are stripped out completely and different forms of a word are reduced to one canonical form. Moreover, every term in a document could be presented as a vector resulting from its frequency in the whole document in relation to terms preceding or following it [15]. The distributional characteristics of the relations between terms can be exploited to generate d-dimensional latent spaces where distance metrics over vectors make it possible to measure document relevance, with a number of dimensions inferior to the number of terms in the corpus: $d < t$.

B. Clustering

Clustering, considered as the prominent task of unsupervised learning, deals with partitioning datasets in meaningful patterns as the basis of further learning steps. There is no clear consensus on the definition of this task, but traditionally, clustering is a procedure that implies [16]: (i) the instances, or data points, in one cluster must be as much as possible similar; (ii) instances of different clusters must be as much different as possible; (iii) dissimilarity (distance) and similarity are basic measures in constructing clustering algorithms [17].

Different strategies can be exploited in partitioning the instances in a dataset [18], [19]. A widely accepted classification frames techniques as:

a) Clustering techniques based on partitioning: Partitioning techniques takes in input a dataset having "n" data points and group them into "k" clusters or partitions. Each cluster contains at least one data point and each data point must belong to a single cluster. The basic idea of this kind of clustering algorithms is to consider the center of a cluster as the best representative point for a partition. K-means [4] and K-medoids [20] are the most widespread representatives of this category. K-means does not clearly define a method for choosing the appropriate number of clusters and highly depend on user choice. Also, k-means does not apply to categorical data.

b) Clustering techniques based on hierarchical structures: The basic idea of this kind of clustering algorithms is to construct the hierarchical relationship among data organizing them in a dendrogram, a diagram representing the distance between clusters and joining or slitting instances based on subsequent distance threshold [6]. There are two general approaches for generating a hierarchical clustering: 1) Agglomerative: initially assumes the points as individual clusters and, at each step, merge the most similar clusters based on a distance function. 2) Divisive: supposes a cluster that contains all data points, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, which cluster to split and how to perform the split.

c) Clustering techniques based on density: The basic idea of this kind of clustering algorithms is that the data which is in a region with a high density of the data space is considered to belong to the same cluster [21]. The typical ones include DBSCAN [5], OPTICS [22] and Mean-shift [23].

Density-Based Spatial Clustering of Application with Noise (DBSCAN) separates data points into three parts. The three parts are core points (points that are within the cluster), Border point (points that are within the neighborhood of the core point), and Noise points (neither core nor border points). It requires the use of the specified minimum radius (Eps) and the minimum number of points required to form a cluster (Min_samples). Although this algorithm deals well with noise, it can not be reliable and shows sensitivity to Min_samples when tested with high dimensional data sets. This algorithm, compared to k-means in terms of creating clustering of varied shapes, discover excellent arbitrary shaped clusters. These methods do not require any predefined number of clusters.

d) Clustering techniques based on fuzzy theory: The basic idea of this kind of clustering algorithms is to change the discrete value of labels, $\{0, 1\}$, which is changed into the continuous interval [0, 1]. In this way, each instance could belong to one or more clusters at the same time. Typical algorithms of this kind of clustering include FCM [24]–[27], FCS [28] and MM [29]. The core idea of FCM is to get membership of each data point to every cluster by optimizing the objective function.

e) Clustering algorithm for large-scale data: Big data has four characteristics, large in volume, rich in variety, high in velocity, and doubt in veracity [30]. The main basic ideas of clustering for big data can be summarized in the following 4 categories: (1) sample clustering [31], [32]; (2) data merged clustering [[8], [33]; (3) dimension-reducing clustering [34], [35]; (4) parallel clustering [36]–[39]. Typical algorithms of this kind of clustering are K-means, BIRCH [8],CLARA [31], CURE [32], DBSCAN [5], DENCLUE [40], Wavecluster [41] and FC [42].

C. Document Clustering

The clustering task has been largely adopted in text mining for leveraging the navigation of a collection of documents [43] or in organizing the outcome response to a user's query from a search engine [44]. These techniques introduced the notion of phrase-based document clustering using a generalized suffix-tree to obtain information about the phrases and to cluster the documents. Suffix naturally organize documents in a hierarchical structure known as lattice, a partially ordered set in which every two elements have a unique superset and a unique subset. [45]. Willett [46] provides a survey on applying hierarchical algorithms into clustering documents.

A new k-means type algorithm for clustering high-dimensional objects in sub-spaces was presented in [47] considering that in high-dimensional data, clusters of objects often exist in sub-spaces rather than in the entire space. Moreover, some methods in text clustering use multiple techniques in parallel. For example, the *Scatter/Gather* [43], a document browsing system based on clustering, uses a hybrid approach involving both K-means and Agglomerative hierarchical clustering. K-means is used because of its efficiency and Agglomerative hierarchical clustering is used because of its quality.

III. A Comparative Study

With this work, we aim to get numerous scientific publications on COVID-19, ingesting these publications by text embedding algorithms, and evaluate different combinations of feature extraction and clustering algorithms. In practice, document clustering often takes the following steps:

A. Tokenization

Tokenization is the process of parsing text data into smaller units (tokens) such as words and phrases. Commonly used tokenization methods include the Bag-of-words model and the N-gram model.

B. Text pre-processing

Some tokens are less important than others. For instance, the most common words, such as “the”, do not help reveal the essential characteristics of a text. So usually it is a good idea to eliminate these words and other characters such as punctuation marks, brackets parenthesis, and double spaces, before doing further analysis. Different tokens might have similar information and we can avoid calculating similar information repeatedly by reducing all tokens to their base form using various stemming and lemmatization dictionaries.

C. Vectorizing Algorithms

The methods for feature extraction play a crucial role in constructing meaningful clusters. After pre-processing the text data, we can then proceed to generate features. Some of the well-known methods are the followings:

1) TfIdf Vectorizer: For document clustering, one of the most common ways to generate features for a document is to calculate the term frequencies of all its tokens, and sometimes it is also useful to weight the term frequencies by the inverse document frequencies. Although not perfect, these frequencies can usually provide some clues about the topic of the document. Besides, the weights of each term based on its inverse document frequency (IDF) in the document collection could add meaningful features. This discounts frequent words with little discriminating power. Finally, to account for documents of different lengths, each document vector is normalized so that it is of unit length.
2) Countvectorizer: The CountVectorizer is a simple approach for both pre-processing a collection of text documents and build a vocabulary of known words (by easily counting the term frequencies), but also to encode new documents using that vocabulary set.
3) Hashing Vectorizer: This method is highly memory efficient since rather than sorting tokens as strings, it encodes them as numerical indexes. This strategy has several advantages: it is very low memory scalable to large datasets as there is no need to store a vocabulary dictionary in memory.
4) Word2Vec: The word2vec algorithm uses a shallow neural network to learn word associated features from a large corpus of text and the output is a set of vectors assigned to each word. This method uses two algorithms: Continuous Bag of Words (CBOW) and Skip-gram [15].
5) Doc2vec is a technique for representing documents as a vector and is a generalization of the word2vec method [48].

D. Clustering Algorithms

a) K-Means: The K-Means algorithm is a two-step procedure:

1) Select K points as the initial centroids.
2) Assign all points to the closest centroid.
3) Recompute the centroid of each cluster.
4) Repeat steps 2 and 3 until the centroids don’t change.

For K-means clustering, the cosine/euclidean measure is used to compute which document centroid is closest to a given document.

b) DBSCAN: It requires two parameters 1) Eps is the starting point and 2) Min_samples is the minimum number of points required to form a dense region. The following steps can elaborate DBSCAN algorithm :

1) An random point is usually taken as the initial point.
2) A parameter Eps is used for determining the neighborhood
3) If there exist sufficient data points or neighborhoods around the initial random point then the algorithm can proceed and this particular data point is labeled as visited or else the point is labeled as a flaw in data or outlier.
4) If this point is considered a part of the cluster then its Eps neighborhood is also the part of the cluster and step 2 is repeated for all Eps. this is repeated until all points in the cluster are determined.
5) Another initial data point is processed and the above steps are restated until all clusters and noise are discovered.

c) MiniBatchKMeans: Mini Batch K-means algorithm‘s main idea is to use small random batches of data of a fixed size, so they can be stored in memory. Each iteration a new random sample from the dataset is obtained and used to update the clusters and this is repeated until convergence. Each mini-batch updates the clusters using a convex combination of the values of the prototypes and the data, applying a learning rate that decreases with the number of iterations. This learning rate is the inverse of the number of data assigned to a cluster during the process. As the number of iterations increases, the effect of new data is reduced, so convergence can be detected when no changes in the clusters occur in several consecutive iterations.

d) BIRCH: BIRCH creates a tree of nodes that summarize data by its accumulated zero, first, and second moments. A node, called Cluster Feature (CF), is a small cluster of numerical data. The construction of a tree in core memory is controlled by some parameters. A new data point descends along the tree to the closest CF leaf. If it fits the leaf well and if the leaf is not overcrowded, CF statistics are incremented for all nodes from the leaf to the root. Otherwise, a new CF is constructed. Since the maximum number of children per node (branching factor) is limited, one or several splits can happen. When the tree reaches the assigned memory size, it is rebuilt and a threshold controlling whether a new point is assigned to a leaf or starts a new leaf is updated to a coarser one. The outliers are sent to the disk and refitted gradually during tree rebuilds.

e) Agglomerative Hierarchial clustering: Agglomerative is an approach of Hierarchical-based clustering. In this algorithm, initially, the points are assumed as individual clusters and, at each step, the most similar clusters will be merged which a definition of cluster similarity.

E. Dimensionality Reduction

Finally, the clustering models should be assessed by various metrics and it is sometimes helpful to visualize the results by plotting the clusters into low dimensional space. To this aim, we use the *t-SNE* algorithm [49] which reduce the returned d-dimensional vectors of embedding algorithm to a $2/3$-dimensional space.

The *t-SNE* algorithm constructs a probability distribution over pairs of high-dimensional vectors in such a way that to similar vectors a higher probability is assigned while dissimilar ones get a lower probability. Then, it defines a similar probability distribution over the vectors in the low-dimensional space and minimizes the *Kullback–Leibler* [50] divergence.

F. Evaluation

Evaluating the performance of a clustering algorithm is not as feasible as counting the number of errors or the precision and recall of a supervised classification algorithm.

The main purpose of evaluation measures in clustering is to test the validity of the algorithm. Evaluation indexes are mainly divided into two groups, the ”internal” and the ”external” one, in terms of the test data whether in the process of constructing the clustering algorithm. The internal evaluation takes the ”internal” data to test the validity of the algorithm. It, however, can’t absolutely judge which algorithm is better when the scores of the two algorithms are not equal based on the internal evaluation indicators [51]. There are three commonly used internal indicators.

1) Silhouette index [9]: The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is (b - a) / max(a, b).
2) Davies_ Bouldin index [10], [11]: This index by measuring the ratio of the sum of within-cluster scatters to between-cluster separations, can identify cluster overlap.
3) calinski_harabasz index [12]: The score is defined as the ratio between the within-cluster dispersion and the between-cluster dispersion.

The external evaluation, which is called the gold standard for the testing method, takes the external data to test the validity of the algorithm. However, it turns out that the external evaluation is not completely correct recently [52]. There are some commonly used methods listed:

1) Entropy: compute the “probability” that a member of cluster j belongs to class i. Then using this class distribution, the entropy of each cluster j is calculated using the standard Shannon entropy formula.
2) F-measure: a measure that combines the precision and recall ideas from information retrieval.
3) Adjusted Rand Index: (ARI) computes a similarity measure between two clustering by considering all possible pairs of samples and counting the pairs that are correctly predicted in-cluster.

IV. Results

In this section, the results of the experimental study are illustrated. As we previously discussed, due to a great variety of approaches in clustering algorithms, we choose K-Mean, DBSCAN, Agglomerative, MiniBatchKmeans, BIRCH models, which are well-known for their accuracy and their applicability in big data. From different vectorizing methods for feature extraction, we choose CountVectorizer, HashingVectorizer, TFIDF Vectorizer, and two of the most recent methods in text embeddings: word2vec and doc2vec.

For evaluating the results we try different indexes due to their applicability and validity in different clustering methods. As an unsupervised learning method, clustering cannot be evaluated using ground truth labels, we then have to focus on intra-cluster properties. In this study, we use three of them, Silhouette, Davies_ Bouldin, and calinski_harabasz indexes, we refer the reader to III-F for a detailed presentation.

For measuring the similarity, initially, two distance measurements, the euclidean and cosine, were considered. The consistency in results convinced us to choose one of these distance measures. Therefore, in illustrating the results just euclidean distance is used in the clustering algorithms.

a) K-Means: The presented results of this method consist of different trials of k, the number of clusters, for reporting Silhouette index.

b) DBSCAN: Using DBSCAN method as an example of density_based algorithms, requires defining the relevant values for Min_samples and radius (Eps). Variations in these two values results in different number of estimated clusters and noise points. The results are depicted in Figures (2, 3, 4, 5 and 6) for various methods of text vectorizing and different values of Min_samples and Eps.

c) Agglomerative, MiniBatchKmeans, BIRCH Clusterings: In this part of experimental results we try more clustering method in our study, document clustering task. We evaluate the efficiency of these methods, based on their achieved Silhouette index. The Figure (7) demonstrating the results for different values of k, the number of clusters.

V. Conclusion

In this paper, we aim to efficiently do the clustering task on the COVID-19 publications. The 27678 number of related publications body-text were used in this study. The results indicated that these publications are highly overlapped in context and mentioned clustering algorithms find difficulties in estimating the efficient number of clusters. In the k-mean algorithm, regardless of the used vectorizing method, we witnessed all the three tested index score, slightly indicate the $k \in [10, 20]$ gets the higher scores, as seen in Figure (1).

In the DBSCAN method, in contrast to k-mean which requires the number of clusters as a predefined parameter, the estimated number of clusters and noises change by variations in Eps and $Min\_samples$. Figures (2,3,4,5,6) are demonstrating the number of estimated clusters and noises in terms of these two parameters. The corresponding results to each vectorizing method, consistently decrease the number of estimated clusters and noises by increasing Eps while increasing the Min_samples leads to an increase in the number of estimated clusters and noises. According to each different vectorizing method, an efficient number of clusters could be assumed for specific Eps and $Min\_samples$ in which the smallest number of noises exist. We purposely didn't use the Silhouette index for DBSCAN due to the poorly estimation of this index for density_based clustering methods. In other experiments, we tried to use other clustering methods, such as Agglomerative, MiniBatchKmeans, and BIRCH, and estimate the evaluation by Silhouette index. The results turn out that the MiniBatchKmeans algorithm clusters the publications more efficiently than the others.

Acknowledgments

This work was partly supported by the European Union's Horizon 2020 research and innovation program under the SMART BEAR project, grant agreement No 857172. It was also partly supported by the program "Piano di sostegno alla ricerca 2019" funded by Universita degli Studi di Milano.

References

[1] A. K. Kar, S. K. Patel, and R. Yadav, "A comparative study & performance evaluation of different clustering techniques in data mining," in *ACEIT Conference Proceeding*, 2016, pp. 139–142.

[2] K. DeFreitas and M. Bernard, "Comparative performance analysis of clustering techniques in educational data mining," *IADIS International Journal on Computer Science & Information Systems*, vol. 10, no. 2, 2015.

[3] C. El Morr and J. Subercaze, "Knowledge management in healthcare," in *Handbook of research on developments in e-health and telemedicine: Technological and social perspectives*. IGI Global, 2010, pp. 490–510.

[4] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[5] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[6] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[7] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 1177–1178.

[8] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: an efficient data clustering method for very large databases," *ACM sigmod record*, vol. 25, no. 2, pp. 103–114, 1996.

[9] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[10] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[11] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of intelligent information systems*, vol. 17, no. 2-3, pp. 107–145, 2001.

[12] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.

[13] J. Ashford, "Gerald kowalski. information retrieval systems," *JOURNAL OF DOCUMENTATION*, vol. 54, pp. 634–635, 1998.

[14] G. Salton, *The SMART Retrieval System—Experiments in Automatic Document Processing*. USA: Prentice-Hall, Inc., 1971.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[16] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[17] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645–678, 2005.

[18] J. M. Kleinberg, "An impossibility theorem for clustering," in *Advances in neural information processing systems*, 2003, pp. 463–470.

[19] A. K. Mann and N. Kaur, "Survey paper on clustering techniques," *International journal of science, engineering and technology research*, vol. 2, no. 4, pp. 803–6, 2013.

[20] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert systems with applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[21] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.

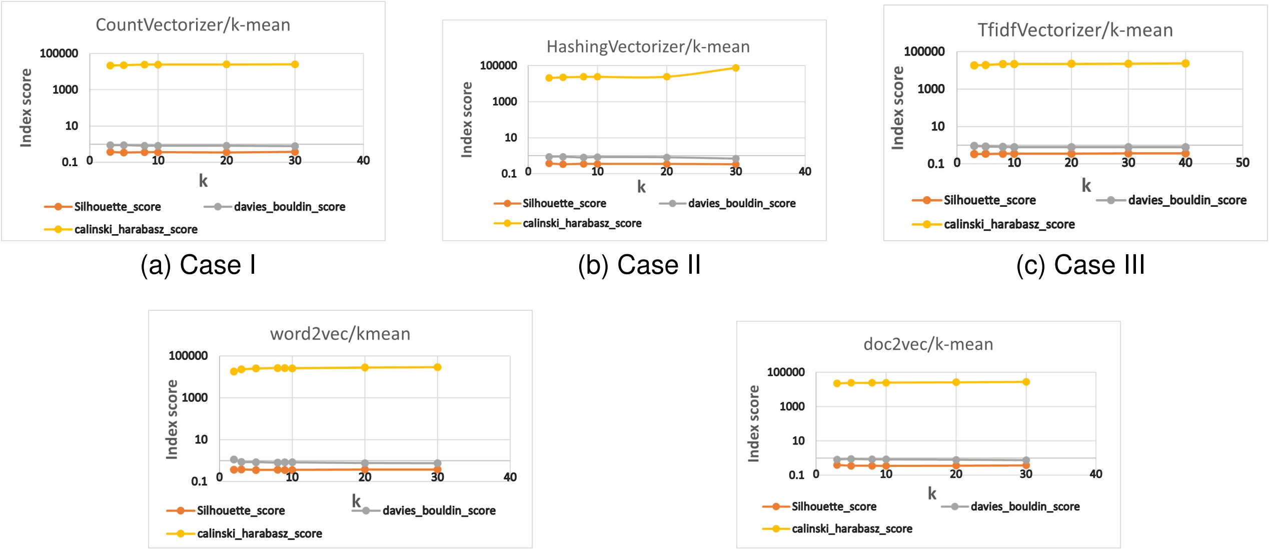
[22] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.

[23] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[24] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," 1973.

[25] J. C. Bezdek, "Objective function clustering," in *Pattern recognition with fuzzy objective function algorithms*. Springer, 1981, pp. 43–93.

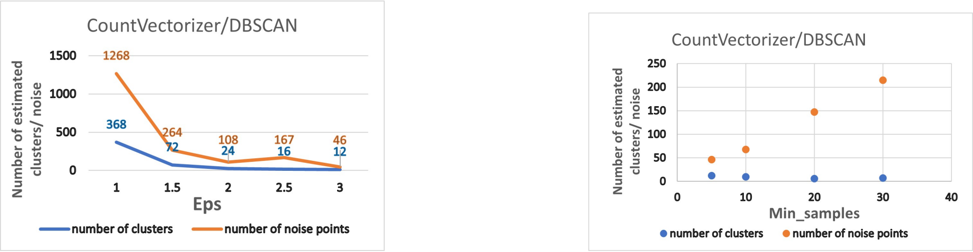
[26] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.



(a) Case I (b) Case II (c) Case III

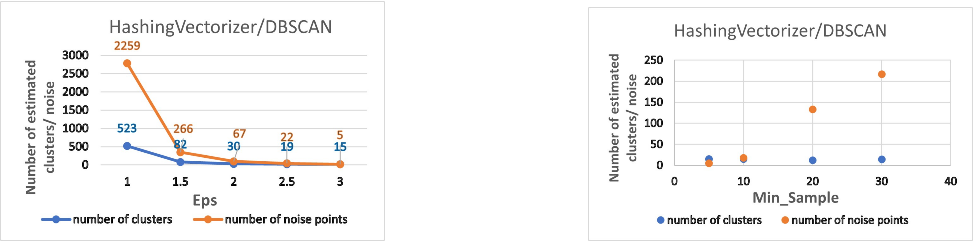
(d) Case IV (e) Case V

Fig. 1. Three index scores, are presented for k-mean clustering methods. In the sub-figs (1a, 1b, 1c, 1d and 1e) respectively, the CountVectorizer, HashingVectorizer, TFIDFVectorizer, word2vec and doc2vec algorithms are used in feature extraction and vectorizing instances.



(a) Case I (b) Case II

Fig. 2. The estimated number of clustering and noise points in term of Eps and Min_samples are plotted rspectively as case I and case II, using the CountVectorizer method in vectorizing and DBSCAN for clustering task.



(a) Case I (b) Case II

Fig. 3. The estimated number of clustering and noise points in term of Eps and Min_samples are plotted rspectively as case I and case II, using the HashingVectorizer method in vectorizing and DBSCAN for clustering task.

[27] P. Ceravolo, E. Damiani, and M. Viviani, "Extending formal concept analysis by fuzzy bags," in *Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, 2006.

[28] R. N. Dave and K. Bhaswan, "Adaptive fuzzy c-shells clustering and detection of ellipses," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 643–662, 1992.

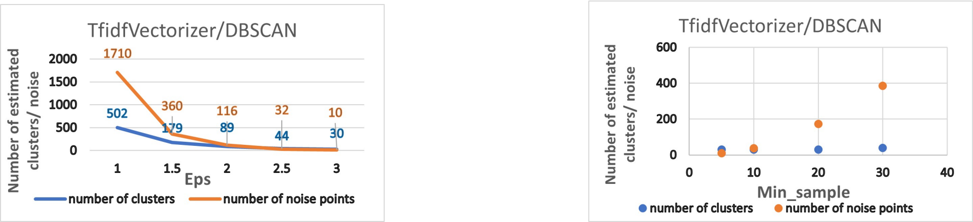
[29] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 8, pp. 1279–1284, 1994.

[30] J. Leskovec, A. Rajaraman, and J. D. Ullman, "Finding similar items," *Mining of Massive Datasets*, pp. 73–130, 2014.

[31] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009, vol. 344.

[32] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," *ACM Sigmod record*, vol. 27, no. 2, pp. 73–84, 1998.

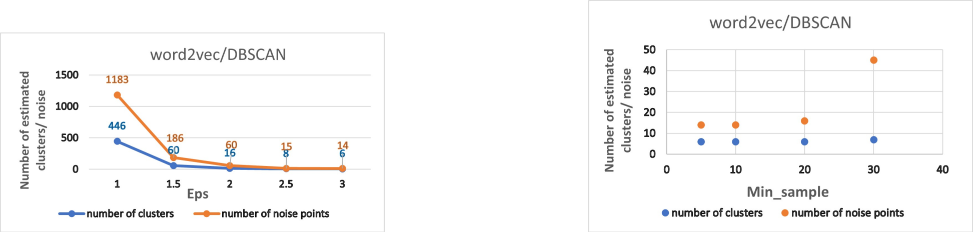
[33] M. S. G. Karypis, V. Kumar, and M. Steinbach, "A comparison of document clustering techniques," in *TextMining Workshop at KDD2000 (May 2000)*, 2000.



(a) Case I

(b) Case II

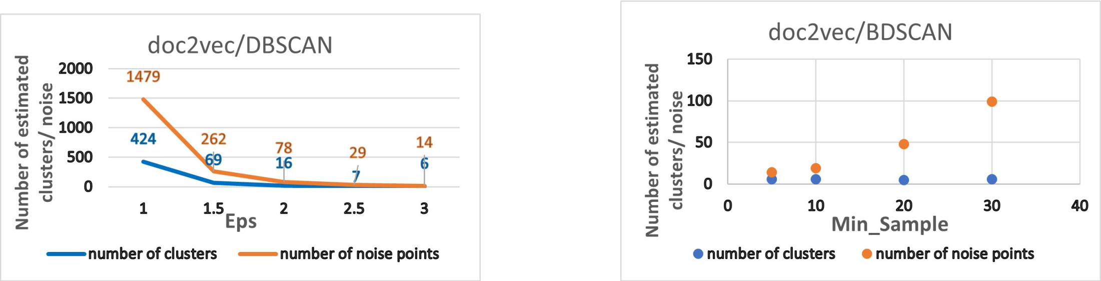
Fig. 4. The estimated number of clustering and noise points in term of Eps and Min_samples are plotted rspectively as case I and case II, using the TFIDFVectorizer method in vectorizing and DBSCAN for clustering task.



(a) Case I

(b) Case II

Fig. 5. The estimated number of clustering and noise points in term of Eps and Min_samples are plotted rspectively as case I and case II, using the word2vec method in vectorizing and DBSCAN for clustering task.



(a) Case I

(b) Case II

Fig. 6. The estimated number of clustering and noise points in term of Eps and Min_samples are plotted rspectively as case I and case II, using the doc2vec method in vectorizing and DBSCAN for clustering task.

[34] L. Parsons, E. Haque, and H. Liu, “Subspace clustering for high dimensional data: a review,” *Acm Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[35] H.-P. Kriegel, P. Kröger, and A. Zimek, “Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 1, pp. 1–58, 2009.

[36] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, “Parallel spectral clustering in distributed systems,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 3, pp. 568–586, 2010.

[37] D. Judd, P. K. McKinley, and A. K. Jain, “Large-scale parallel data clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 871–876, 1998.

[38] D. K. Tasoulis and M. N. Vrahatis, “Unsupervised distributed clustering.” in *Parallel and distributed computing and networks*, 2004, pp. 347–351.

[39] W. Zhao, H. Ma, and Q. He, “Parallel k-means clustering based on mapreduce,” in *IEEE international conference on cloud computing*. Springer, 2009, pp. 674–679.

[40] A. Hinneburg, D. A. Keim *et al.*, “An efficient approach to clustering in large multimedia databases with noise,” 1998.

[41] G. Sheikholeslami, S. Chatterjee, and A. Zhang, “Wavecluster: A multi-resolution clustering approach for very large spatial databases,” in *VLDB*, vol. 98, 1998, pp. 428–439.

[42] D. Barbará and P. Chen, “Using the fractal dimension to cluster datasets,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 260–264.

[43] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, “Scatter/gather: A cluster-based approach to browsing large document collections,” in *ACM SIGIR Forum*, vol. 51, no. 2. ACM New York, NY, USA, 2017, pp. 148–159.

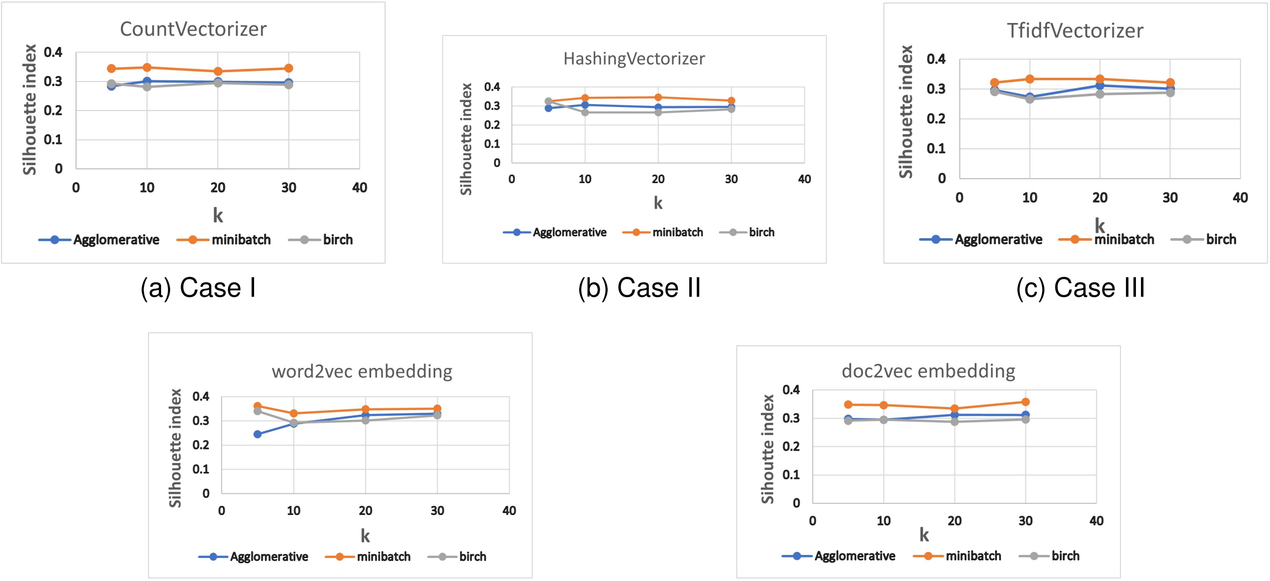
[44] O. Zamir, O. Etzioni, O. Madani, and R. M. Karp, “Fast and intuitive clustering of web documents.” in *KDD*, vol. 97, 1997, pp. 287–290.

[45] D. Koller and M. Sahami, “Hierarchically classifying documents using very few words,” Stanford InfoLab, Tech. Rep., 1997.

[46] P. Willett, “Recent trends in hierarchic document clustering: a critical review,” *Information processing & management*, vol. 24, no. 5, pp. 577–597, 1988.

[47] L. Jing, M. K. Ng, and J. Z. Huang, “An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data,” *IEEE Transactions on knowledge and data engineering*, vol. 19, no. 8, pp. 1026–1041, 2007.

[48] Q. Le and T. Mikolov, “Distributed representations of sentences and



(a) Case I

(b) Case II

(c) Case III

(d) Case IV

(e) Case V

Fig. 7. The comparative Silhouette indexs for five vectorizing algorithms and three different clustering methods, are plotted.

documents," in *International conference on machine learning*, 2014, pp. 1188–1196.

[49] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *Advances in neural information processing systems*, 2003, pp. 857–864.

[50] S. Kullback and R. Leibler, "10.1214/aoms/1177729694," *Ann. Math. Stat*, vol. 22, pp. 79–86, 1951.

[51] V. Estivill-Castro, "Why so many clustering algorithms: a position paper," *ACM SIGKDD explorations newsletter*, vol. 4, no. 1, pp. 65–75, 2002.

[52] I. Färber, S. Günnemann, H.-P. Kriegel, P. Kröger, E. Müller, E. Schubert, T. Seidl, and A. Zimek, "On using class-labels in evaluation of clusterings," in *MultiClust: 1st international workshop on discovering, summarizing and using multiple clusterings held in conjunction with KDD*, 2010, p. 1.