Progress in Artificial Intelligence (2019) 8:123–132
https://doi.org/10.1007/s13748-018-0165-5

REGULAR PAPER

Fuzzy clustering-based semi-supervised approach for outlier detection in big text data

Farek Lazhar[1]

Received: 13 April 2018 / Accepted: 2 September 2018 / Published online: 14 September 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract
Text data is often polluted by outlier documents which can significantly influence the performance of classification techniques. In this paper, we propose an approach based on fuzzy clustering to detect outlier documents. The principle of our approach is based on the assumption that documents assigned to different clusters with very close degrees are considered as candidate outliers. Firstly, a semantic data model is built using Doc2Vec framework. Secondly, a fuzzy clustering is performed. Thirdly, candidate outlier documents are detected based on the different degrees of membership. Finally, for each candidate outlier, the objective function is recomputed, and a candidate document is considered as outlier when it conducts to considerably increase the objective function score. To show the effectiveness of our approach, two classification tests, one with original datasets and the second without outlier, are applied. Experimental results show that discarding outlier from datasets conducts to improve the performance of classifiers.

Keywords Outlier detection · Fuzzy clustering · Big text data · Doc2Vec modeling · Sparsity · High dimensionality · Classification

1 Introduction

Outlier detection, also called anomaly detection, is the process that identifies divergent observations, i.e., observations which are not strongly related to the majority of observations in the same dataset. Outlier is defined as the set of objects that are considerably dissimilar from the remainder of the data [1]. Outlier is generally a data point which is different from the normal behavior of data points [2], also defined as a data value that seems to be out of place with respect to the rest of data [3]. Due to its effectiveness in data-mining area, outlier detection has been widely studied and attracted much attention of researchers in several domains including defense, fraud detection, and agriculture. Various researches have been conducted on the outlier detection and their application in various domains [2–4].

Clustering also called cluster analysis is an efficient and effective data-mining process for aggregating a set of objects into clusters (partitions) according to their similarities. Clustering aims to find a structure that aggregates the data into some groups with the property that data belonging to a group (or cluster) are more similar to data in that cluster than to data in other clusters [5]. Two types of clustering can be distinguished: crisp clustering also called hard clustering where each observation is affected completely to only one cluster. In contrast, fuzzy clustering also called soft clustering, each observation is assigned to all clusters with belonging degrees ranging from 0.0 to 1.0. However, different researchers employ different cluster models, and for each of these cluster models, again different algorithms can be given [6].

The problem of clustering can be very useful in the text domain, where the objects to be clusters can be of different granularities such as documents, paragraphs, sentences or terms [7]. However, big text data, as its name indicates, it is characterized by its high dimensionality where redundant and irrelevant features are often present. The conducted study [8] proved the inefficiency for measuring proximity in high dimensionality by showing surprising behavior of distance metrics.

The importance of this study is to show the effect of outlier documents on the behavior of classification techniques, i.e., showing the effect of removing outlier documents on the classification accuracy where data which are charac-

✉ Farek Lazhar
farek.lazhar@univ-guelma.dz; fareklazhar@gmail.com

[1] University of Guelma, BP 411, Guelma, Algeria

terized by both its sparsity and high dimensionality form the key issue raised by our approach and characterizes the points of difference with respect to methods that deal with numerical data in terms of outlier detection. To deal with the problem of sparsity and high dimensionality, topic modeling techniques are used such as Latent Dirichlet Allocation (LDA), Latent Semantic Analysis, Nonnegative Matrix Factorization (NNMF) and recently Doc2Vec which proved its effectiveness in capturing better semantic similarity between documents.

In text mining, outlier documents carry much noise and make distance far from discriminative documents; hence, outlier documents are considered as misleading for a classifier because of the high level of ambiguity they carry, which finally decrease its performance. In our work, and in order to make sense to distance measurement when applying a fuzzy clustering process, Doc2Vec framework is applied.

To show the effectiveness of fuzzy clustering on outlier detection for big text data, in this paper, we propose a fuzzy clustering-based approach for detecting outlier documents. The remainder of this paper is organized as follows. Section 2 presents a literature review on some related works. In Sect. 3, the proposed approach is presented. Experimental datasets are presented in Sect. 4. Empirical results are discussed in Sect. 5. Our research work is concluded in Sect. 6.

2 Literature review

Outlier detection is widely studied and attracted the attention of several researchers. Due to the vagueness of outlier techniques, in this section, we review some general methods using some different techniques that deal with outlier detection and other related methods focusing specifically on clustering techniques.

Rousseeuw and Leroy [9] reviewed regression techniques used to identify outlier in numerical datasets. Identifying outlier data points using either simple or multiple regression is based on an optimization technique called least square (LS), where a data point represented by its explanatory and response variables is considered as outlier when it deviates from the linear relation. For this purpose, a robust fit of data is used and outliers are those points having large residuals from the robust equation.

A distance-based algorithm [10] based on large dataset partition has been designed for outlier detection. Deciding either a point is outlier or not is based on computing the distance from its kth nearest neighbor, and then, top n points are considered as outliers. Firstly, candidate partitions are generated using BIRCH's pre-clustering algorithm [11]. Secondly, an algorithm called nested-loop algorithm is used for computing outliers from the candidate partitions. Empirical results show its efficiency in outlier detection with respect to both data set size and data set dimensionality outperforming the nested-loop algorithm and another one called index-based algorithm.

In the same context, a distance-based approach [12] has been proposed for detecting outlier in large high-dimensional datasets. Similar to the proposed algorithm [10], a designed algorithm called HilOut is designed to efficiently detect the top n outliers where the sum of the distances separating a data point from its k nearest-neighbors is used as a weighting scheme. Data are linearized using the notion of space-filling curve. Two functions called temporal cost and special cost, respectively, are used in the first phase to provide an approximate solution. In this phase, the algorithm iteratively isolates candidate data points to be outliers and at once reduces the dataset size. The algorithm stops when the dataset size reaches n. In the second phrase, an exact solution is provided examining further the candidate outliers that remained after the first phase. Tested on large high-dimensional datasets, the proposed algorithm always reports good solutions after a finite number of iterations.

Clustering methods can be categorized into four main categories: partitioning clustering, hierarchical clustering, density-based clustering and grid-based clustering [1,13].

K-means algorithm and its variants, such as K-medoids, K-medians, K-modes, fuzzy K-means [14], are widely used in the literature of partitioning methods. Numerous other variants of these algorithms such as Balanced Iterative Reducing and Clustering Hierarchies (BIRCH) [11], Partitioning Around Medoids (PAM) [15], Clustering LARge Applications (CLARA) [16] and Clustering Large Applications based upon RANdomized Search (CLARANS) [16] are also used. These algorithms are applied on data streams for outlier detection, and combining some of them shows more effectiveness in detecting outlier than using separated ones. PAM clustering algorithm [15] has been applied considering that small-sized clusters are good holders for outlier objects. An algorithm called I-CLARANS [17] has been proposed, which is indeed a modified variant of CLARANS algorithm [16] using some geometric proprieties and that to identify outlier. Empirical results show that I-CLARANS algorithm performs better in detecting outlier compared to PAM , CLARA and CLARANS.

Hierarchical methods involve creating clusters using either bottom-up or top-down strategy. Bottom-up strategy aims to merge cluster into larger cluster until all objects are into one cluster or until some conditions for termination are satisfied. In contrast, top-down strategy aims to subdivide the cluster into small clusters. Numerous algorithms such as BIRCH [11] and Cluster Using Representatives (CURE) [18] are used. BIRCH algorithm [11] has been applied on a large dataset for detecting outlier, and empirical results show that BIRCH [11] outperforms CURE and another algorithm called CHAMELEON [19].

In density-based methods [20], a data point is declared as outlier if its local density is low with respect to the remaining data points. Some proposed approaches [10,21] consider that a data point is outlier if it is situated far from the dense regions of data. Density-Based Spatial Clustering of Application with Noise (DBSCAN) [22] is the most well-known density-based clustering algorithm, used to discover cluster of arbitrary shapes. It was used to detect anomaly in temperature data [22], compared to statistical anomaly detection methods which detect only extreme values, and they find that DBSCAN is able to find other values which are not necessary extreme.

Grid-based clustering algorithms are efficient in mining large multidimensional data sets [23], and a grid-based clustering method takes a space-driven approach by partitioning the embedding space into cells independent of the distribution of the input objects. Algorithms such as Statistical Information Grid approach (STING) [24] and Clustering in Quest (CLIQUE) [5] are used.

A distance-based algorithm called ODDC (Distribution Clustering Outlier Detection) [25] has been proposed for outlier detection. It is based on mapping data into a new feature space, where the transformation vector captures the distance distribution of each point. ODDC proved its efficiency compared to a well-known outlier detection algorithm called LOF (Identifying Density-Based Local Outliers) [26] regard to the size of datasets, the dimensionality and the percentage of outliers.

A combined approach proposed by Gath and Geva [27] using fuzzy clustering with maximum-likelihood estimation (MLE) has been developed for modeling of finite mixtures of normal distributions and accurately estimates the underlying parameters. Empirical results showed the robustness of the proposed algorithm against convergence to singularities and its high speed of convergence.

Based on fuzzy K-means algorithm, a method [28] has been proposed for outlier detection, where underlying parameters have been estimated using the fuzzy performance measures. Based on the hypothesis that data can be homogenous or not, outliers relative to a unimodal distribution can be detected for both univariate and multivariate data and that by transforming the original data from n-dimensions to one dimension using a jackknife procedure. Observations with large residuals should be then labeled as outliers.

3 Proposed approach

Focused on fuzzy clustering, the core idea of our approach is based on the assumption that data points assigned to clusters with very close belonging degrees are supposed to be more deviating than data points biased toward a specific cluster. Since clustering is based on the mean-squared distortion as the objective function, it should be useful to employ this function to test the deviation of candidate data points.

To transform brut textual data from sparse high-dimensional space to a compact vector space, Doc2Vec framework is used and that to capture semantic similarity between representative vectors when applying the fuzzy clustering.

3.1 Problem overview

Given a corpus of n documents $D = \{d_1, d_2, \ldots, d_n\}$, where each document d_i is assigned to one target class from $C = \{c_1, c_2, \ldots, c_k\}$, k is the number of classes.

Let cl a classification technique. The goal of this work is to detect a subset S from D that poorly affects the performance of cl. By removing S, the accuracy of cl should be performed compared to its performance before removing S, i.e., discarding outlier documents from D can help to improve the accuracy of cl.

By converting our classification problem into a fuzzy clustering problem with k clusters, each document assignment will be distributed over the k clusters. A document is considered as a candidate outlier if it is situated near to clusters boundary, i.e., the probability of belonging is approximately equally distributed among all clusters. Then, iteratively, an objective function score $\hat{J}$ is recomputed for each candidate document; if $\hat{J}$ will be increased, the candidate document is considered as an outlier.

Figure 1 shows the mains steps of our approach.

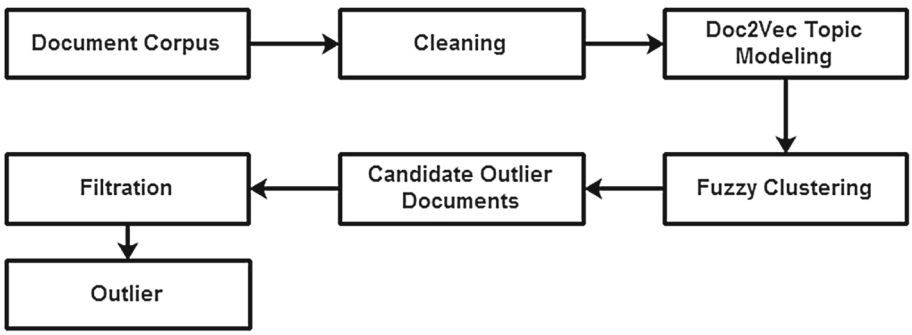

Fig. 1 Overview of the proposed approach

3.2 Data cleaning

Text data may carry much irrelevant and undesirable contents such as stop words, punctuation, and HTML tags; hence, for an effective content, cleaning text data is very recommended to reduce the sensitivity of clustering algorithms to the high dimensionality. Knowing that an important number of features can poorly affect the clustering results in terms of computing similarity distance between documents.

Stop words are words that are frequent and do not carry any meaning, including articles, prepositions and some other high-frequency words, such as 'a', 'the', 'of', 'and', 'and', 'it', 'I', 'you', 'that', 'this' and 'those'. Punctuation and special chars are also high-frequent and do not carry any meaning such as ;, !, ?, %, >, # and &.

HTML tags are keywords surrounded by angle brackets like <html>, <head>, and , which do not carry semantic information. By removing those frequent tags, the dimension of the corpus should be considerably reduced.

3.3 Topic modeling with Doc2Vec

Text data is characterized by its high dimensionality and sparsity which poorly affect the clustering where distance measurement is a main component. Distance measures like the Euclidean distance for high-dimensional data exhibit surprising properties that differ from what is usual for low-dimensional data [29]. To overcome the problem of sparsity and dimensionality, semantic modeling methods are used.

Doc2Vec, also Paragraph Vector is one of the most effective data modeling techniques to learn document-level embedding and represent documents as a vector, introduced in 2014 by Le and Mikolov [16], which is in fact a generalizing of Word2Vec method [30], and that by extending the learning of embedding from words to word sequences. Word2Vec is a three-layer neural net with one input, one hidden and an output layer. It implements Continuous Bag of Words (CBOW) and SkipGram architectures for computing vector representations of words, including their context [31]. CBOW tries to predict a word on bases of its neighbors, i.e., predicting a word given its context (syntactic relation). However, SkipGram tries to predict the neighbors of a word, i.e., predicting the context given a word (sematic relation).

Formally, it is described as follows: Every word is mapped to a unique vector, represented by a column in a matrix W. Given a sequence of training words $w_1, w_2, \ldots, w_T$, the objective of the word vector model is to maximize the average log probability [16]. Given a word w and its surrounding (context) words, CBOW and SkipGram, respectively, optimize the following objective functions :

$$\zeta_{\text{CBOW}} = \sum_{w \in W} \log \quad p\left(w \mid \text{Context}\left(w\right)\right) \tag{1}$$

$$\zeta_{\text{SkipGram}} = \sum_{w \in W} \log \quad p\left(\text{Context}\left(w\right) \mid w\right) \tag{2}$$

Doc2Vec explores Word2Doc framework by adding additional input nodes representing documents as additional context. Each additional node can be thought of just as an identifier for each input document.

The objective of Doc2Vec learning is:

$$\max \sum \log \quad p\left(\text{tar} \mid (\text{con}, \text{doc})\right) \tag{3}$$

where tar: target word, con: context words, doc: document context.

In our study, choosing Doc2Vec for topic modeling is motivated by the empirical evaluation [32] which shows its effectiveness when it is trained on large corpora compared to other embedding methodologies. Also, the comparative study of semantic modeling methods [31] shows that Doc2Vec outperformed other semantic modeling methods such as LSA and LDA.

3.4 Fuzzy *K*-means algorithm

Clustering is often used to group the documents, in the hope that each group will represent documents with a common theme or topic [33]. Fuzzy clustering algorithms have been proved to be a better method than hard clustering in dealing with discrimination of similar structure [34].

Fuzzy K-means also known as soft K-means or fuzzy C-means is an extension of K-Means, it was introduced by Bezdek [14] in 1981, and it is based on the fuzzy set theory [35] which assign observations to more than one cluster with variable belonging degrees that can vary from 0 to 1. Fuzzy K-means strategy is based on minimizing the sum of squared error (SSE) objective function, defined as follows:

$$J = \sum_{i=1}^{m} \sum_{j=1}^{C} \mu_{ij}^{m} \left\| x_j - cj \right\|^2 \tag{4}$$

where m is a fuzzification coefficient greater than 1, μ_{ij} is the degree of membership of x_i in the cluster j, x_i is a data point, c_j is the center of the cluster j and $\|*\|$ is a distance function to measure the similarity between any data point x_i and the center c_j.

Fuzzy partitioning is carried out through an iterative optimization of the objective function (Eq. 4), with the update of membership μ_{ij} and the cluster centers c_j by:

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \tag{5}$$

$$c_j = \frac{\sum_{i=1}^{N} \mu_{ij}^m x_i}{\sum_{i=1}^{N} \mu_{ij}^m} \tag{6}$$

This iteration stops when $\max_{ij} \left| \left\{ \mu_{ij}^{(k)} - \mu_{ij}^{(k+1)} \right\} \right| < \epsilon$, where ϵ is a termination criterion between 0 and 1, whereas k are the iteration steps. This procedure converges to a local minimum or a saddle point of J. The algorithm is illustrated as follows.

Algorithm 1: Fuzzy K-Means

Input : Initial centroids $c_i = (i = 1, \cdots k)$, • Initial $U = [u_{ij}]$ matrix $(U^{(0)})$, • $X = [x_1, x_2, \ldots, x_n]$: data, • ϵ: threshold value(stop criterion)

Output : C: updated centroid matrix

repeat

calculate the centers vectors:$C^{(k)} = [c_j]$ with $U^{(k)}$

$$c_j = \frac{\sum_{i=1}^{N} \mu_{ij}^m x_i}{\sum_{i=1}^{N} \mu_{ij}^m}$$

Update the membership matrix $U^{(k+1)}$

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

until $max_{ij} \left| \left\{ \mu_{ij}^{(k)} - \mu_{ij}^{(k+1)} \right\} \right| < \epsilon$;

return C

In our work, since we combine supervised classification techniques, we propose that the chosen number of clusters is the number of classes (i.e., number of topics).

3.5 Similarity measurement

Similarity is one of the key issues of cluster analysis, which means that one of the most influential elements of cluster analysis is the choice of an appropriate similarity measure [36]. The similarity between objects is defined by a distance measure, which plays an important role in obtaining correct clusters [37]. Choosing a proper technique for distance calculation is totally dependent on the type of the data that we are going to cluster [38]. Distance similarity plays a crucial role in clustering [37]. It defines how the similarity of two elements (x, y) is calculated and it will influence the shape of the clusters. In most high-dimensional applications, the choice of the distance metric is not obvious, and the notion for the calculation of similarity is very heuristical [8].

In this subsection, we briefly present the most commonly used measures [39,40] to quantify the similarity between data points.

Euclidean distance The Euclidean distance, also known as L2 norm, between two data points x_i and x_j is defined as:

$$d_{\text{euc}}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{7}$$

Manhattan distance Manhattan distance, also known as a city block, rectilinear or L1 distance, is mathematically defined as:

$$d_{\text{man}}(x, y) = \sum_{i=1}^{n} |x_i - y_i| \tag{8}$$

Cosine distance Cosine distance is a measure of similarity between two vectors by measuring the cosine of the angle between them. It is defined as:

$$d_{\cos}(x, y) = 1 - \left(\frac{x_i^T y_i}{\|x_i\| \, \|y_i\|} \right) \tag{9}$$

Other distance measures are also used such as the correlation distance and Pearson correlation coefficient [39–41].

Since the problem of high dimensionality and sparsity of text data can be solved using Doc2Vec topic modeling, we assume that the standard Euclidean distance is largely enough for measuring distance between vectors representing documents.

3.6 Candidate outlier detection

The data model obtained from cluster detection may be very sensitive to outliers. The outliers could be considered as "rare"; data, not following the overall tendencies of the group [42].

In our study, since each document is assigned to all clusters with different belonging degrees, hence, a document is biased toward the cluster that carry the highest degree and it is far from the clustering boundary. However, documents near to the boundary which carry very close belonging degrees are considered as candidate outliers.

Consider a data point x_1 assigned to three clusters c_1, c_2, c_3 with the belonging degrees 0.7500, 0.1200 and 0.1300, respectively, and x_2 another data point assigned to the same clusters with the belonging degrees 0.3310, 0.3315 and 0.3375, respectively. We can say that x_1 is biased to c_1 with a degree of 0.7500, i.e., x_1 has a chance of 75% to be a member of c_1. However, x_2 belonging degrees are very close which make the decision very ambiguous; hence, x_2 is considered as a candidate outlier.

Formally, consider D a set of documents, clustered into k fuzzy clusters: $C = c_1, \ldots, c_k$. Let COD a candidate set of

outlier documents. A document from COD should verify the following inequality:

$$\left|\mu_{ci}(d) - \mu_{cj}(d)\right| \leq t \tag{10}$$

where c_i and c_j are two fuzzy clusters from C and $\mu_{ci}(d)$ and $\mu_{cj}(d)$ are the belonging degrees of d in c_i and c_j, respectively.

The following algorithm shows how to detect candidate outliers using belonging degrees:

Algorithm 2: Candidate Outlier Detection

Input : $D = d_1, d_2, \ldots, d_n$, $C = c_1, c_2, \ldots, c_k$, t : a user fixed threshold
Output : COD: Candidate Outlier Documents

```
for each document d in D do
    recover all belonging degrees of d:
    P = {μ_c1(d), μ_c2(d), ..., μ_ck(d)}
    for all combinations (μ_ci(d), μ_cj(d)) in P do
        if |μ_ci(d) − μ_cj(d)| ≤ t then
            COD = COD ∪ d
        end
    end
end
```

Note that the user threshold t varies from 0 to 1, if $t = 0$ that means belonging degrees are equally distributed over all the clusters and only points situated on the cluster boundary are considered as candidates, a small value of t greater than 0 implies that points situated near to the edge are considered as candidates, if $t = 1$ all data points are candidates.

3.7 Candidate outlier filtration

After extraction of candidate outlier set, for each candidate document represented by its Doc2Vec vector, the fuzzy K-Means algorithm is re-executed to compute the new score of the objective function.

Now, we have for each candidate outlier document its corresponding score from the objective function. Each point where its score deviates positively from the rest of scores is considered as an outlier. Note that a positive deviation of a score causes an increase in the objective function (i.e., increase the sum of squared error (SSE)).

Since our candidate data points are represented by numerical values (i.e., scores if the objective function), and to measure the spread (variability) of scores, a common statistical method using the standard deviation is applied to identify distant points (documents vectors) that are further away from the mean. Let m and δ the mean and the standard deviation of scores, respectively. For N candidate data points, m and δ are given by the following formulas:

$$m = \frac{1}{N}\sum_{i=1}^{N}\hat{J}(i) \tag{11}$$

$$\delta = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{J}(i) - m\right)^2} \tag{12}$$

Standard deviation-based outlier detection method remove points that are above $(m + 2 \times \delta)$ and points that are below $(m - 2 \times \delta)$. In this work, we take into account only candidate points that increase the objective function; hence, each data point d within a score $\hat{J}$ that verify the following inequality is considered as an outlier:

$$\hat{J}(d) > m + 2 \times \delta \tag{13}$$

The following algorithm shows the filtration process.

Algorithm 3: Candidate Outlier Filtration

Input : COD: Candidate Outlier Documents (Doc2Vec vectors), S: a set of Objective Function Scores
Output : OD: Outlier Documents

```
for each document d in COD do
    Re-execute Fuzzy K-Means clustering
    Compute Objective Function Score: Ĵ(d)
    S = S ∪ Ĵ(d)
end
N = |S|
m = (1/N) Σ_{i=1}^{N} S(i)
δ = sqrt((1/N) Σ_{i=1}^{N} (S(i) − m)^2)
for each document d in COD do
    if Ĵ(d) > m + 2 × δ then
        OD = OD ∪ d
    end
end
return OD
```

4 Experimental datasets

Three datasets have been used to evaluate the performance of our approach. These datasets are widely used text-mining area. The following table shows the total number of documents and classes for each dataset.

Reuters-R8 Extracted from Reuters-21578 dataset, Reuters-R8 dataset contains 7674 labeled documents using eight classes: acq, crude, earn, grain, interest, money-fx, ship, trade.

20NewsGroups A well-known dataset, it contains approximately 20 000 press articles labeled among 20 classes.

C50 Also called reuters-50-50, it consists of texts from Reuters Corpus Volume 1 (RCV1). The C50 corpus has 50 authors with documents that belongs to CCAT category

(about corporate in industrial news), and each author has 50 documents to train and 50 documents to test [43] which give in total 5000 documents labeled within 50 categories.

In our study, we used the Reuters-R8 dataset in full (7674 documents labeled within 8 classes). Only the first 10 classes are used for 20NewsGroups and C50 datasets which give for each dataset 11,314 and 1000 documents, respectively.

5 Experimental results

Using datasets mentioned in Sect. 4. The first phase is dedicated to test the ability of our approach in mining outlier documents where data cleaning, Doc2Vec modeling, fuzzy K-means clustering are accomplished. In the second phase, classifiers are tested. At first, with original datasets (i.e., with outlier documents) and the performance in terms of $F1$-measure is computed. Then, the same classifiers without outlier documents are applied to show the effect of outlier documents on the behavior of classification algorithms.

In the second phase, three popular classification techniques commonly used in text classification are used: naive Bayes (NB), support vector machine (SVM), stochastic gradient descent classifier and (SGD classifier). A detailed description of these three techniques can be found in [44]. We evaluated the performance of chosen classifiers with the $F1$-score which is given by the following formula:

$$F1 = 2 \times \frac{p \times r}{p + r} \tag{14}$$

where Precision $p = \text{tp}/(\text{tp}+\text{fp})$ is the fraction of all positive predictions that are actual positives. Recall $r = \text{tp}/(\text{tp} + \text{fn})$ is the fraction of all actual positives that are predicted to be positive.

True positives tp, false positives fp, false negatives fn and true negatives tn are represented via a confusion matrix (Table 1).

Candidate outlier documents are identified using Algorithm 2. Note that a best user threshold value of t is fixed to 0.25 and that after a manual test of some values ranging from 0.0 to 0.5. Then, candidates are filtered using Algorithm 3, and points that obviously deviate from the mean are considered as outliers.

Figures 2, 3 and 4 show for each dataset (Table 2), fuzzy clusters, candidate outliers and objective function scores corresponding to outlier documents. Red points indicate scores of outlier points (documents) that obviously increased the objective function.

Table 3 shows the results of classification before and after outlier detection.

Table 4 shows for each dataset the total number of documents, number of cadidate outliers and number of detected outliers.

As shown in Table 3, despite the slight improvement recorded after outlier removal, we can say that outlier documents have a negative influence on classifications algorithms when comparing $F1$-measure before outlier removal.

Although our approach is time-consuming when the number of candidate outliers is important, the challenging problems of high dimensionality and sparsity have been overcome using Doc2Vec topic modeling framework and that to make useful the use of the distance metrics when calculating similarity between centers of clusters and representative Doc2Vec vectors.

To highlight the difficulties when dealing with large textual data and in order to compare our approach in terms of using fuzzy logic with respect to the nature of data, we discuss below different ways of exploiting fuzzy K-means capabilities by some approaches to obtain satisfactory results.

Fuzzy K-means proved its efficiency when dealing with quantitative data. For example, in [27], fuzzy K-means algorithm combined with maximum-likelihood estimation (MLE) applied on normal univariate and bivariate datasets showed that fuzzy clustering proved its convenience for the estimation of the underlying parameters.

However, in case of high-dimensional data, a transformation process of feature space is needed to show its efficiency. In [28], after transforming data sets using a jackknife procedure, results show that fuzzy K-means algorithm played an important role where degrees of membership have been used to calculate a performance measure called fuzzy hypervolume F. Outlying observations are then detected by performing a standard test of significance carried out on the values of F using the mean and the standard deviation parameters.

In our work, transforming categorical data into numerical one using Doc2Vec modeling framework was intended to make sense to distance metrics when measuring similarity between representative vectors. This transformation of data can be seen as an important process which decreased the difficulty of working with textual data as part of outlier detection regard to other approaches that work only with quantitative normally (Gaussian) distributed datasets. We can cite as examples, the research work [27] where test datasets are quantitative and supposed normally distributed. The same case in [28] where data are also quantitative and unimodal distributed (i.e., with only one peak), which is not the case for our work, where data are categorical, unstructured, sparse

Table 1 Confusion matrix

	Actual positive	Actual negative
Predicted positive	tp	fp
Predicted negative	fn	tn

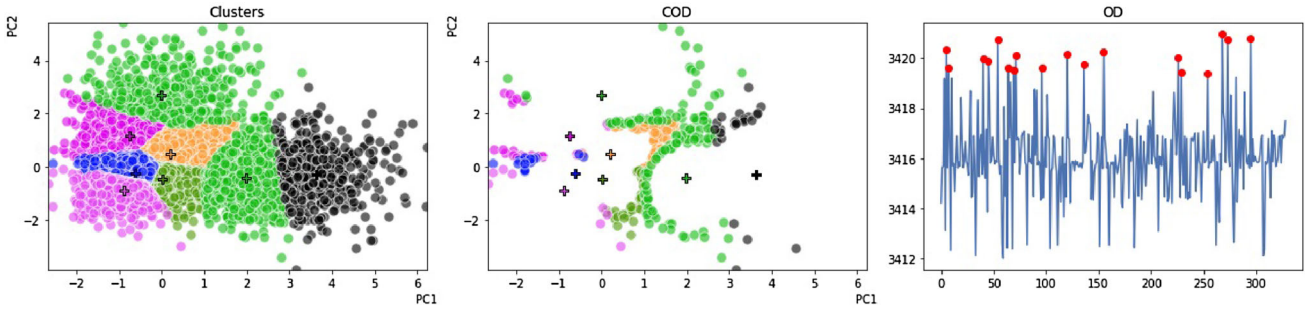

Fig. 2 Reuters-R8

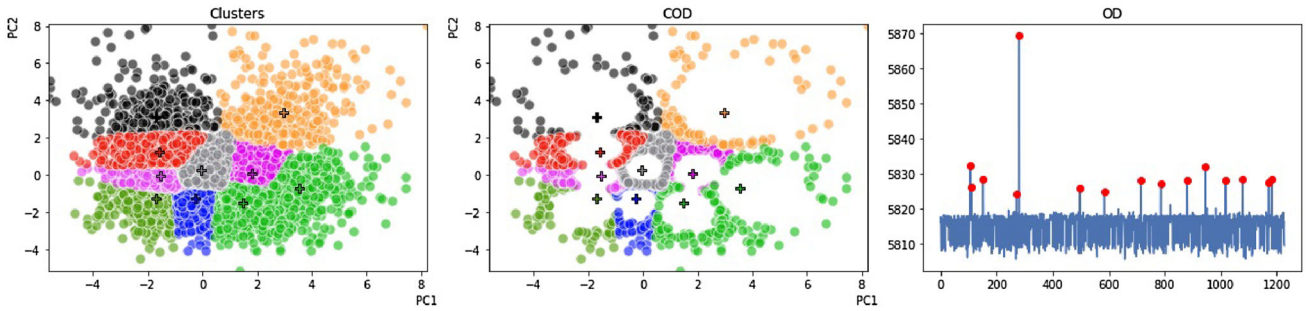

Fig. 3 20NewsGroups

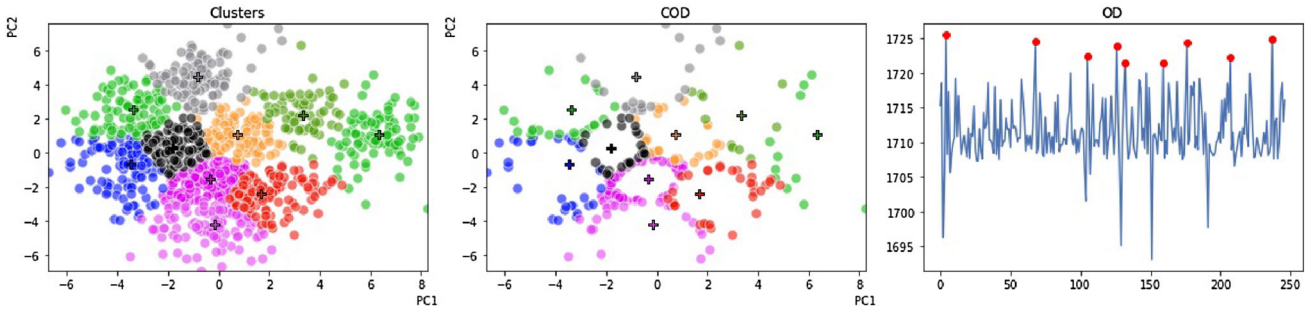

Fig. 4 C50

Table 2 Experimental datasets

Dataset	#Documents	#Classes
Reuters-R8	7674	8
20NewsGroups	20,000	20
C50	5000	50

and high-dimensional and therefore cannot be statistically consistent.

Also, for normal distributed data, several transformation techniques can be applied. In [12], dataset is transformed using space-filling curve as a linearization technique which make sense to the distance between data points which then helped to detect outlier points. Very similar to [12], the distance-based approach [25] shows that for each data point computing distances distributed over clusters can help to detect outliers. This method transits the feature space to the new space by discretizing the distance distribution of each object where each point in the original space is represented as a new vector, and each dimension of the new vector is a dis-

Table 3 Classification results before and after outlier removal

Dataset	Classifier	$F1$	
		Before	After
Reuters-R8	NB	0.920521	0.920941
	SVC	0.990228	0.990228
	SGD	0.987622	0.989577
20NewsGroups	NB	0.871330	0.884283
	SVC	0.911054	0.918826
	SGD	0.914508	0.919689
C50	NB	0.742000	0.759000
	SVC	0.846000	0.855000
	SGD	0.840000	0.872000

Table 4 Total number of documents, number of cadidate outliers and number of detected outliers

Dataset	#Documents		
	Total	Candidates	Outliers
Reuters-R8	7674	329	18
20NewsGroups	11,314	1128	15
C50	1000	471	9

tance distribution. After clustering, objects in small clusters are considered as outliers.

However, these methods of transformation cannot be applied to textual data, where distance measurements proved the inefficiency when measuring semantic similarity between documents. The problem has been resolved by using Doc2Vec framework which eliminates the issue of sparsity and high dimensionality; therefore, capturing semantic similarity became possible. Subsequently, as the performance of fuzzy K-means depends on the appropriate use of distance metrics, it is now possible to capture semantic similarities between documents which will finally lead to do meaningful statistical analysis.

6 Conclusion

In this paper, we proposed a semi-supervised approach based on fuzzy K-means clustering algorithm in order to detect outlier documents in big text data. We faced two main challenging problems in the text- mining area which are high dimensionality and sparsity. Using Doc2Vec topic modeling framework to represent data into a semantic vector space and reduce dimensionality, the distance calculation between vectors when applying fuzzy K-means has become useful.

According to the belonging degrees of data points across the different clusters, a data point is considered as a candidate outlier if belonging degrees are very close. For each candidate outlier, and objective function score is computed. Then, each candidate represented by its score is considered as an outlier when it deviates considerably from the mean of all candidate points scores.

As a future study, we intend to apply other variants of fuzzy clustering algorithms within different distance metrics to investigate the best method for detecting outlier documents.

References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, vol. 743. Morgan Kaufmann, San Francisco (2006)
2. Tamboli, J., Shukla, M.: A survey of outlier detection algorithms for data streams. In: 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp 3535–3540 (2016)
3. Sreevidya, S.S.: A survey on outlier detection methods. Int. J. Comput. Sci. Inf. Technol. **5**(6), 8153–8156 (2014)
4. Sharma, S., Jain, R.: Outlier detection in agriculture domain: application and techniques. In: Aggarwal, V., Bhatnagar, V., Mishra, D. (eds.) Big Data Analytics. Advances in Intelligent Systems and Computing, vol. 654. Springer, Singapor (2018)
5. Assent, I.: Efficient density-based subspace clustering in high dimensions. In: Masulli, F., Petrosino, A., Rovetta, S. (eds.) Clustering High-Dimensional Data. Lecture Notes in Computer Science, vol. 7627, pp. 34–49. Springer, Berlin (2015)
6. Merrell, R., Diaz, D.: Comparison of data mining methods on different applications: clustering and classification methods. Inf Sci Lett Lect Notes Comput Sci **4**(2), 61–66 (2015)
7. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceeding COLT' 98 Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 92–100 (1998)
8. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (eds.) Database Theory ICDT 2001. Lecture Notes in Computer Science, vol. 1973. Springer, Berlin (2001)
9. Rousseeuw, P., Leroy, A.: Robust Regression and Outlier Detection, 3rd edn. Wiley, New York (1996)
10. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: SIGMOD Conference, pp. 427–438 (2000)
11. Jagadeeswaran, V.S., Uma, P.: Detection of noise by efficient hierarchical BIRCH algorithm for large data sets. Int. J. Adv. Res. Comput. Commun. Eng. **2**(2), 1306–1309 (2013)
12. Angiulli, F., Pizzuti, C.: Outlier mining in large high-dimensional data sets. IEEE Trans. Knowl. Data Eng. **17**(2), 203–215 (2005)
13. Jain, A.K., Murty, M.N., Flyn, P.J.: Data clustering: a review. ACM Comput Surv **31**(3), 264–323 (1999)
14. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
15. Kumar, V., Kumar, S., Singh, A.K.: Outlier detection: a clustering-based approach. Int. J. Sci. Mod. Eng. **1**(7), 16–19 (2013)
16. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: ICML'14 Proceedings of the 31st International Conference on International Conference on Machine Learning, Beijing, China, vol. 32, pp. II-1188–II-1196 (2014)
17. Singh, G., Kumar, V.: An efficient clustering and distance based approach for outlier detection. Int. J. Comput. Trends Technol. **4**(7), 2067–2072 (2013)
18. Guha, S., Rastogi, R., Shim, K.: CURE: an efficient clustering algorithm for large databases. In: ACM SIGMOD Conference, vol. 27(2) (1998)
19. Karypis, G., Han, E.H., Kumar, V.: Chameleon: hierarchical clustering using dynamic modeling. IEEE Comput. **32**(8), 68–75 (1999)
20. Breunig, M.M., Kriegel, H.P. ,Ng, R.T., Lof, S.J.: Identifying density-based local outliers. In: SIGMOD Conference, pp. 93–104 (2000)
21. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: Proceeding VLDB Algorithms for Mining Distance-Based Outliers in Large Datasets, pp. 392–403 (1998)
22. Çelik, M., Dadaşer-Çelik, F., Dokuz, A.Ş.: Anomaly detection in temperature data using DBSCAN algorithm. In: International Symposium on Innovations in Intelligent Systems and Applications (INISTA), Istanbul, Turkey, pp. 91–95 (2011)
23. Mirkin, B.G.: Clustering for Data Mining: A Data Recovery Approach, vol. 3. CRC Press, Boca Raton (2005)

24. Wang, W., Yang, J., Muntz, R.: STING: a statistical information grid approach to spatial data mining. In: Proceedings of the 23rd International Conference on Very Large Data Bases, pp. 186–195. Morgan Kaufmann Publishers Inc., Burlington (1997)
25. Niu, K., Huang, C., Zhang, S., Chen, J.: ODDC: outlier detection using distance distribution clustering. In: Washio, T. (ed.) PAKDD 2007 Workshops. Lecture Notes in Artificial Intelligence (LNAI), vol. 4819, pp. 332–343. Springer, Berlin (2007)
26. Breunig, M.M., Kriegel, H., Ng, R.T., et al.: LOF: identifying density-based local outliers. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Dalles, TX, pp. 93–104 (2000)
27. Gath, I., Geva, A.: Fuzzy clustering for the estimation of the parameters of the components of mixtures of normal distribution. Pattern Recognit. Lett. **9**, 77–86 (1989)
28. Cutsem, B., Gath, I.: Detection of outliers and robust estimation using fuzzy clustering. Comput. Stat. Data Anal. **15**, 47–61 (1993)
29. Klawonn, K., Höppner, F., Shim, K., Jayaram, B.: Efficient algorithms for mining outliers from large data sets. In: Proceeding Revised Selected Papers of the First International Workshop on Clustering High-Dimensional Data, vol. 7627, pp. 14–33 (2013)
30. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at the International Conference on Learning Representations, Scottsdale, USA (2013)
31. Campr, M., Ježek, K.: Comparing semantic models for evaluating automatic document summarization. In: Král, P., Matoušek, V. (eds.) Text, Speech, and Dialogue. TSD 2015. Lecture Notes in Computer Science, vol. 9302. Springer, Cham (2015)
32. Lau, J.H., Baldwin, T.: An empirical evaluation of doc2vec with practical insights into document embedding generation. In: Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, Germany, pp. 78–86 (2015)
33. Ertöz, L., Steinbach, M., Kumar, V.: Finding topics in collections of documents: a shared nearest neighbor approach. In: Ertöz, L., Steinbach, M., Kumar, V. (eds.) Clustering and Information Retrieval. Network Theory and Applications, vol. 11. Springer, Boston (2004)
34. Bayley, M.J., Gillet, V.J., Willett, P., Bradshaw, J., Green, D.V.S.: Computational analysis of molecular diversity for drug discovery. In: Proceeding of the 3rd Annual Conference on Research in Computational Molecular Biology, pp 321–330. ACM Press, New York (1999)
35. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
36. Sami, Ä., Tommi K.: Introduction to partitioning-based clustering methods with a robust example. Reports of the Department of Mathematical Information Technology, University of Jyväskylä, Finland (2006)
37. Bora, D.J.: Computational analysis of molecular diversity for drug discovery. Int. J. Comput. Sci. Inf. Technol. **5**(2), 2501–2506 (2014)
38. Bora, D.J., Gupta, A.K.: Effect of different distance measures on the performance of K-means algorithm: an experimental study in Matlab. Int. J. Comput. Sci. Inf. Technol. **5**(2), 2501–2506 (2014)
39. Kull, M., Flach, P.A.: Reliability maps: a tool to enhance probability estimates and improve classification accuracy. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2014. Lecture Notes in Computer Science, vol. 8725. Springer, Berlin (2014)
40. Wang, F., Sun, J.: Survey on distance metric learning and dimensionality reduction in data mining. Data Min. Knowl. Discov. **29**(2), 534–564 (2015)
41. Wu, W.: Clustering and information retrieval. In: Feature Selection for High-Dimensional Data. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer, Cham (2015)
42. Wen, J.R., Zhang, H.J.: Query clustering in the web context. In: Wu, W., Xiong, H., Shekhar, S. (eds.) Clustering and Information Retrieval. Network Theory and Applications, vol. 11. Springer, Boston (2004)
43. López-Monroy, A.P., Montes-y-Gómez, M., Villaseñor-Pineda, L., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: A new document author representation for authorship attribution. In: Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Olvera López, J.A., Boyer, K.L. (eds.) Pattern Recognition. MCPR 2012. Lecture Notes in Computer Science, vol. 7329. Springer, Berlin (2012)
44. Forsyth, D.: Learning to classify. In: Probability and Statistics for Computer Science. Springer, Cham (2018)