Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

Normalized Target Feature Projective Regression Based Bootstrap Aggregative Document Clustering In Cloud

Mrs B. BalaVinothini[1] , Dr. N.Gnanambigai [2], Dr. P. Dinadayalan[3]

[1] Bharathiar University, Coimbatore 641046, Tamil Nadu, India

[2] Department of Computer Sci., Indira Gandhi College of Arts and Science, 605009 Puducherry, India.

[3] Department of Computer Sci., Kanchi Mamunivar Government Institute for Post graduate Studies and Research, Puducherry, India.

Abstract: Text document clustering is the process of separating particular documents' sets into varied groups based on certain similarity criteria in different areas of text mining for attaining better precision or recall in the retrieval systems. Due to the large volume of data with high dimensionality in cloud storage, relevant document retrieval is the challenging one. Therefore, document clustering is used to retrieve more relevant collection of documents to the user's query. A novel Normalized Target Feature Projective Regression-based Bootstrap Aggregative Document Clustering (NTFPR-BADC) technique is introduced for retrieving the more relevant documents with higher accuracy and lesser computational complexity. The proposed NTFPR-BADC technique comprises three processes, initially in preprocessing words are removed, then in feature extraction process target projective pursuit regression is utilized, and in document clustering bootstrap aggregative ensemble technique is applied to provide the final clustering results with higher accuracy. These clustered document is uploaded to the cloud storage and relevant document retrieved based on the user query. The comprehensive experimental assessment is carried out using different factors such as precision, recall, and computational complexity with respect to the number of documents collected from the dataset. By these results the proposed NTFPR-BADC technique achieving higher precision, recall, and lesser computational complexity than the existing methods.

Keywords: Cloud, preprocessing, target projective pursuit regression feature extraction, bagging ensemble clustering, document retrieval.

1. Introduction

Clustering is generally used to extract the valuable patterns in a dataset. This clustering is also often applied to a text document to facilitate the process of various systems, such as image recognition, text classification, information retrieval, and so on. In the cloud, a large volume of text is generated every day. This huge amount of mostly unstructured text documents not easy to process. Therefore, efficient and effective methods and algorithms are required to find out useful patterns. Text mining is the process of extracting significant information and it has significant attention in recent years. In this paper, an ensemble clustering algorithm is developed for essential text mining tasks and it includes text document pre-processing, feature extraction, and clustering. In-text mining, many algorithms have been introduced in the existing literature with the aim of solving the clustering problems with the large data stored in the cloud.

An improved spherical k-means clustering algorithm was designed in [1] for partitioning a large number of documents. The designed algorithm was not accurately clustering documents with minimum time. Also, the precision and recall rate of the algorithm was not increased. Advanced Document Clustering (ADC) was designed in [2] to partition the documents with higher accuracy using cosine similarity. The designed ADC technique failed to employ a projection scheme to produce smaller dimensional features for further improving the clustering performance and also guarantee computational efficiency. A hybrid Krill herd (KH) algorithm was introduced in [3] with the objective of grouping the text documents. Though the algorithm increases the performance of precision and recall, the complexity was not minimized.

An automated consensus clustering technique was introduced in [4] to partition the documents into different clusters. But the designed technique failed to provide the dimensionality reduced features. A concept factorization (CF) with adaptive neighbors based clustering technique was presented in [5] to execute the dimensionality reduction. But the higher clustering performance was not achieved.

An unsupervised feature transformation (UFT) was developed in [6] for extracting the more similar features and also performing the document categorization using the fuzzy clustering technique. The designed approach was not efficient to obtain the fuzzy clustering and dimension reduction.

Memetic Differential Evolution algorithm was designed in [7] to perform the text categorization. However, the algorithm failed to concentrate on implementing different validity measures. A particle swarm optimization algorithm was introduced in [8] to select the feature for creating a new subset and also increasing the performance of the text categorization with lesser computational time. However, an efficient clustering technique was not used to achieve higher true positive and lesser false positives.

A deep-learning vocabulary network was presented in [9] to collect and arrange the text documents into significant clusters for mining the usage patterns and also achieving a better clustering performance. But the dimension reduction was not performed while processing a large number of text documents. A Singular Value Decomposition–based clustering technique was presented in [10] to create various clusters. The designed technique has higher complexity for creating a hierarchy of clusters.

A modified genetic algorithm was designed in [11] for categorizing the text documents on the cloud. However, the deigned algorithm was not efficient for handling the huge amount of documents. A Length Feature Weight (LFW) for the vectorized representation used to partition the documents in [12]. But the complexity analysis was not performed. A probabilistic based Latent Dirichlet Allocation model was developed in [13] to considerably increase the accuracy of document clustering. The model failed to extract significant features for increasing clustering performance.

A hierarchical Dirichlet multinomial allocation (HDMA) scheme was designed in [14] for increasing the performance of multi-source document clustering. The designed scheme did not accurately reduce the false positives rate while clustering the numerous documents. A Spectral Clustering algorithm was presented in [15] for partitioning the text documents based on the particle swarm optimization. However, the required minimization of the computational complexity was not achieved.

A fuzzy clustering framework was developed in [16] for document categorization based on the cosine-distance similarity. But, the feature selection was not performed. A scalable hierarchical clustering framework was introduced in [17] based on input documents pre processing and feature vectors transformation. However, the accurate clustering algorithm was not applied to perform new document categorization.

A two-stage sentence selection approach was developed in [18] to partition the documents. However, the model was not efficient to partition the documents with higher precision and recall. A distributed shared nearest neighbor (D-SNN) algorithm was designed in [19] to partitions the text documents. The algorithm failed to implement with the big data for increasing the clustering quality.

A Fuzzy Bag-of-Words (FBoW) technique was developed in [20] for document representation by measuring the similarity between the words. But, the computation complexity of document representation was not reduced.

1.1 Our contributions

The existing issues are reviewed from the above sections are overcome by introducing a novel technique called NTFPR-BADC. The contributions of NTFPR-BADC are stated as follows,

- To improve the accuracy of document clustering, a novel NTFPR-BADC is introduced based on different processing steps namely preprocessing, feature extraction, and clustering.
- The collected documents from the dataset are applied to the processing steps to obtain the words and remove the stop and stem words. The preprocessing of the documents increases performance clustering.
- A target projective pursuit regression is applied to analyze the words based on the frequency measure. The occurrences of the word at many times are considered as a keyword and it is called target features. These features are projected and info the feature space and are used for reducing dimensionality.

- The Bootstrap aggregative voting based document clustering technique is applied in NTFPR-BADC to partition the documents based on the extracted keywords. The ensemble technique combines the clustering results of the weak hypothesis and it makes strong clustering results. The clustered results are stored into the cloud and the user retrieves similar documents. As a result, the precision and recall rate is increased.
- Finally, a qualitative analysis is carried out with the various related clustering algorithms to discover the performance improvement of the NTFPR-BADC technique with different performance metrics.

1.2 Organization of the paper

The rest of this article is further organized into four various sections as follows. Section 2 introduces a proposal NTFPR-BADC methodology for the text document clustering. Section 3 illustrates the experimental settings with parameter descriptions. Section 4 provides the various results analysis of the proposed techniques with the help of the table and graphical representation. At last, section 5 concludes the paper.

2. Proposal Methodology

The proposed NTFPR-BADC technique is introduced for clustering the text documents for efficient relevant information retrieval with minimum time. In the cloud, a large number of documents related to the different applications are presented. In this case, a relevant document related to the user query retrieval with minimum time is the most challenging one. Therefore, processing, feature extraction, and clustering are an essential process to find more relevant documents according to the user query. The proposed NTFPR-BADC technique initially performs pre processing and feature extraction for minimizing the dimensionality of the data set before the document clustering.

The overall architecture of the proposed NTFPR-BADC technique is given architecture below. Figure 1 illustrates an architecture diagram of the proposed NTFPR-BADC technique for clustering the text documents.

Initially, the documents are collected from the dataset. After the document collection, the pre processing is performed.

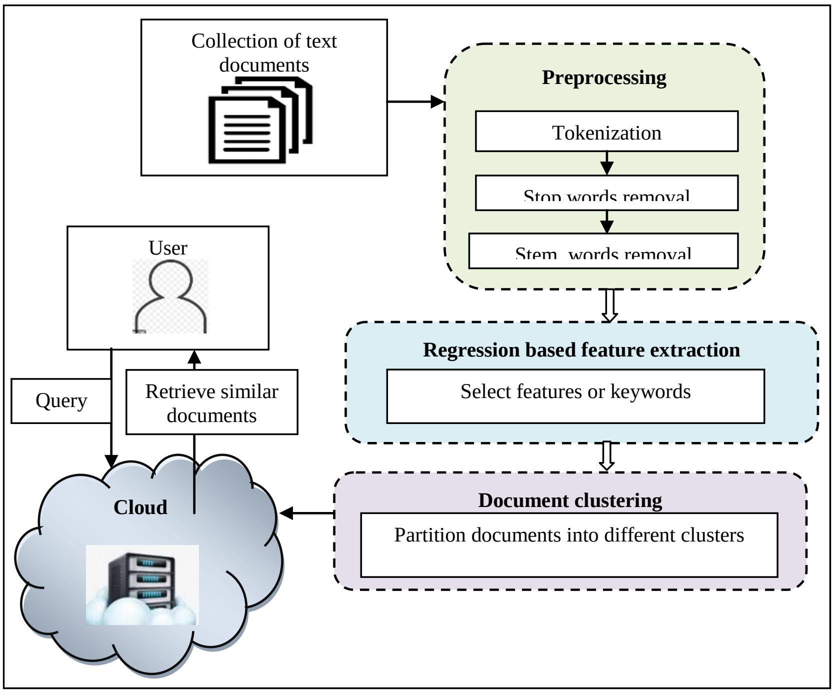
Collection of text documents
Preprocessing
Tokenization
Stop words removal
Stem words removal
User
Query
Retrieve similar documents
Regression based feature extraction
Select features or keywords
Cloud
Document clustering
Partition documents into different clusters

Fig. 1 architecture of the proposed NTFPR-BADC technique

In the second step, the target projective pursuit regression is applied to extract the keywords or features. Finally, the bootstrap aggregative document clustering is performed to group the documents with the extracted features. The detailed process of the proposed NTFPR-BADC technique is explained in the following subsections.

2.1 Normalized text document pre processing

The proposed NTFPR-BADC technique starts to perform the pre processing which includes three different sub-processes namely tokenization, stop word removal, and stem word removal. Initially, the numbers of raw documents are collected from the dataset

$$V = \{v_1, v_2, \dots, v_n\} \in D \quad ...(1)$$

From (1),V denotes a document set $\{v_1, v_2, \dots, v_n\}$, D denotes a dataset. After collecting the documents from the dataset, a set of the input documents are undergone the normalized text document preprocessing. It is the process of changing the text documents into a structured form.

Initially, the tokenization process is performed for segmenting the text into words in the form of part of speech which includes nouns, verbs, and adjectives.

$$v_i = w_1, w_2, w_3, \dots w_n \quad \dots(2)$$

Where, v_i denotes a documents split into the words $w_1, w_2, w_3, \dots w_n$.

Then stop words are removed from the extracted words. Stop words are the words that occur continually in the documents and it did not provide any meaning. The certain stop words are “the”, “a”, “an”, “in, “and”, “our”, “this” and so on. These words are removed from the given documents.

The word stem is a part of a word that provides slightly different meanings. The stem word removal is the process of decreasing the words from their root word. In other words, the word stemming removes the suffixes and provides the root words. The example of the stem words removal is shown in below.

Table 1 example of stem word removal

Word	Removal	Root word
Standing	ing	Stand
Logically	ly	Logic
Ended	ed	End

For example, if the word ends with ‘ing’, ‘ly’, 'ed', are removed within the documents and obtain the root word stand, logic and end.

Algorithm 1. Normalized text document preprocessing

// Algorithm 1 Normalized text document preprocessing

```
Input: dataset D, number of documents $v_1, v_2, \dots, v_n$,
Output: Pre processed documents
Begin
  1. Collect a number of documents $v_1, v_2, \dots, v_n$ from D
  2.   For each document 'v'
  3.     Perform tokenization to segment the text into words $w_1, w_2, w_3, \dots w_n$
  4.     Perform pre processing to remove the stop and stem words
  5.   End for
End
```

Algorithm 1 given above describes the document pre processing to minimize the complexity of the clustering. The tokenization process segments the texts into a number of words. Then the stop and stem words are removed in the given documents.

2.2 Target projective pursuit Regression-based feature extraction

After the pre processing, the keywords (i.e. features) extraction process is carried out to reduce the dimensionality. Target projective pursuit Regression (TPPR) is a machine learning technique that attempts to analyze the given input. Here the target is said to be a relevant feature that is projected into subsets.

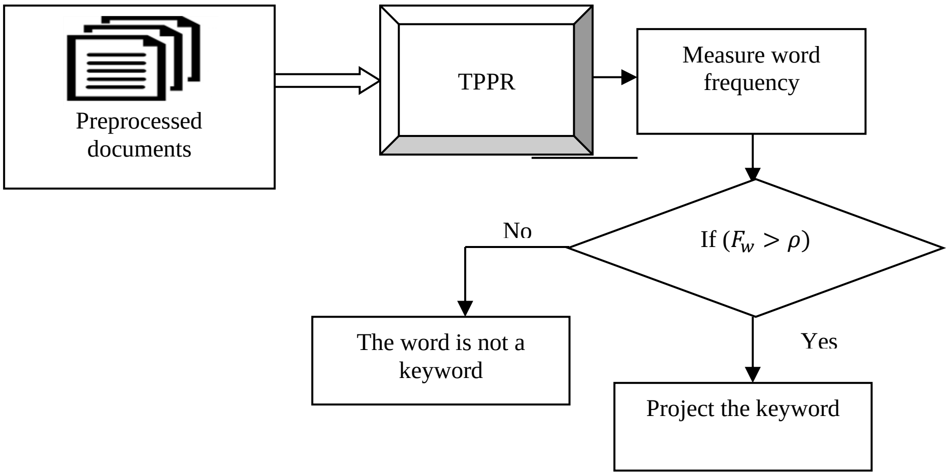

Fig. 2 block diagram of the Target projective pursuit Regression

Let us consider the pre processed documents 'PD', each document consists of a number of words $w_1, w_2, w_3, \dots w_n$. Among the number of words, the keywords (i.e. features) $F_1, F_2, F_3, \dots F_m$ are extracted from the documents for accurate clustering. The TPPR is applied for analyzing the words and identify the keywords based on the term frequency measure. In other words, the words repeatedly occurred in the document are selected as keywords. Therefore, the word frequency is estimated as given below,

$$F_w = \left(\frac{NT_w\,(D)}{T_w(D)}\right) \quad \dots(3)$$

Where, F_w denotes a word frequency, $NT_w\,(D)$ denotes the number of times the words appear in the document, '$T_w(D)$' indicates a total number of words in the document. Then the threshold is set to the term frequency for identifying the keywords.

$$Y = \begin{cases} F_w > \rho\,, \ \text{keyword} \\ \text{otherwise, not a keyword} \end{cases} \quad \dots(4)$$

Where, Y denotes an output of regression function, F_w denotes a word frequency, ρ indicates a threshold. The frequency of the word (F_w) is greater than the threshold (ρ) is selected as a keyword (i.e. features). Otherwise, the word is not a keyword. The selected features are called target and project the keywords for further processing as a result it reduces the dimensionality and time consumption for clustering the documents.

Algorithm 2 Target projective pursuit Regression-based feature extraction

// Algorithm 2 Target projective pursuit Regression-based feature extraction
Input: Pre processed documents PD, **Output**: Dimensionality-reduced feature extraction

```
Begin
    For each preprocessed document 'PD'
        Analyze the words $w_1, w_2, w_3, \dots w_n$
        For each word $w_i$
            Measure word frequency '$F_w$'
            If $(F_w > \rho)$ then
                Word is said to be a keyword
            else
                Word is said to be not a keyword
            End if
        End for
        Project the keywords
    End for
```

Algorithm 2 given above illustrates the process of regression-based feature extraction to minimize the dimensionality. With the application of preprocessed documents, the target projective pursuit regression is applied to analyze the words. The word frequency is measured to identify the repeated occurrence of words. Then the threshold is set and to verify the word frequency. If the frequency of the word is greater than the threshold, then the word is said to be a keyword. Otherwise, the word is not a keyword. Then the regression projects the features as target keywords. As a result, the computational complexity is reduced.

2.3 Bootstrap aggregative voting based document clustering

After the feature extraction, the document clustering is performed using the Bootstrap aggregative voting technique. Bootstrap aggregating, also known as bagging, is a machine learning technique to improve the accuracy of clustering than the boosting technique.

A Bootstrap aggregating ensemble uses the weak hypothesis to partition the documents into different clusters. The weak hypothesis is a base clustering technique that is only slightly correlated with the true clustering results. In contrast, an ensemble technique is a strong clustering technique that provides accurate results.

Figure 3 demonstrates the flow process of the bagging ensemble clustering technique for categorizing the text document with higher accuracy. Let us consider training sets as $(\mathrm{D_i}, \mathrm{Y_i})$. Here $\mathrm{D_i}$ represents a document collection $\{v_1, v_2, \ldots, v_n\}$ i.e. input extracted features 'EF' and $\mathrm{Y_i}$ indicates clustering results for the given inputs.

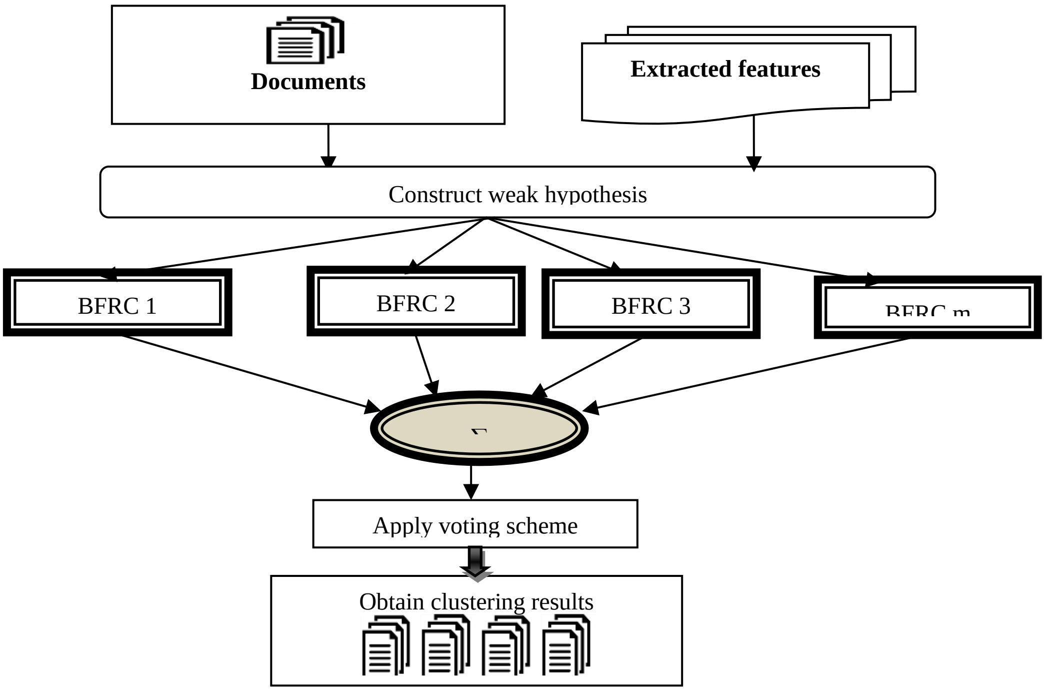

Fig. 3 flow process of Bootstrap aggregative voting

The bagging technique constructs' weak hypothesis to cluster the given input documents. The bagging technique uses the BFR clustering technique as a weak hypothesis. By applying the BFR clustering technique, 'k' number of clusters and centroid (i.e. mean) are initialized based on the extracted features. Then for each cluster, mean and variance is measured as given below,

$$\mathrm{R} = \frac{1}{\beta\sqrt{2\pi}} \exp\left[-0.5\left(\frac{\mathrm{D_i}-\mathrm{m}}{\beta}\right)^2\right] \quad ...(5)$$

Where R denotes an output of BFR clustering technique, β denotes a standard deviation, m indicates a mean, $\mathrm{D_i}$ symbolize the documents. The mean and deviation for a cluster may be dissimilar for different dimensions. The document which is closer to the mean is grouped into the particular cluster. Similarly, all the documents are grouped into different clusters. The weak hypothesis did not provide accurate clustering results. Therefore, the entire weak clustering results are combined and to make a strong one. The strong clustering results are attained as follows,

$$S = \sum_{i=1}^{m} R_i\ (D) \quad ...(6)$$

From (6), ‘S’ indicates strong clustering results, and R_i (D) symbolizes the output of the weak hypothesis. By applying the voting method, the majority votes of the samples i.e. clustering results for each weak hypothesis are correctly identified.

$$F = \arg\max_{m} \varphi\ (D) \quad ...(7)$$

Where F denotes voting output, arg max denotes an argument of the maximum function to find the majority vote (φ) of the samples (i.e.D) whose decision is known to the m^{th} weak hypothesis. In other words, the results having majority votes are considered as final clustering results. In this way, all the documents are correctly partitioned and it is uploaded to the cloud server.

Whenever the user sends the query to the cloud server, the cloud server sends the response from the clustered documents with minimum time. As a result, the user retrieves the more related documents from the cloud server with higher accuracy.

Algorithm 3 bootstrap aggregative document clustering

// Algorithm 3 bootstrap aggregative document clustering
Input: Number of documents $v_1, v_2, \dots, v_n$, Extracted features ‘EF’ **Output**: Document clustering and Retrieved documents

```
Begin
For each document ‘$v_i$’ with selected features or keywords ‘EF’
    Construct ‘m’ number of weak hypothesis
    Initialize the clusters and cluster mean
    for each mean $m_i$
        for each document ‘$v_i$’
            Measure ‘R’
            If document closer to mean ‘$m_i$’ then
                Group documents into particular clusters
            End if
            Obtain weak hypothesis results ‘R’
        end for
    end for
    Combine all weak hypothesis $S = \sum_{i=1}^{m} R_i (D)$
    Identify the majority votes of the samples $F = \arg\max_{m} \varphi (D)$
    Obtain strong clustering results
Upload documents to a cloud server
The user sends a query to the server
Retrieve the relevant documents
```

The above algorithmic step illustrates the step by step process of the document clustering. The bagging ensemble technique considers the documents with the extracted features. The ensemble clustering technique constructs the number of weak hypotheses to cluster the documents based on the extracted features. The outputs of the weak hypothesis are combined and apply the voting scheme. The majority votes of the documents are obtained as final clustering results. As a result, the proposed technique improves the precision and recall rate as well as minimizes the false positive rate.

3. Experimental Settings

The Experimental evaluation of the proposed NTFPR-BADC technique and existing improved spherical k-means clustering algorithm [1], ADC [2] are implemented using Java language in the cloud environment. For the implementation process, the Cranfield collection dataset is used and it is taken from http://ir.dcs.gla.ac.uk/resources/test_collections/cran/. The dataset comprises the 1400 collection of text documents to perform the clustering. Initially, the documents were split into different words. Then the stop and stem words are removed. After the pre processing, the major keywords are extracted based on the frequency measure. Based on the extracted keywords, the documents are clustered. For the implementation, 200 documents are considered with ten iterations.

3.1 Parameter Description

The quantitative analysis of the proposed NTFPR-BADC technique and existing improved spherical k-means clustering algorithm [1], ADC [2] are discussed with different parameters such as recall rate, precision rate, and computational complexity.

The recall rate is measured as the ratio of relevant documents that are correctly retrieved to the total number of relevant documents. Therefore, the recall rate is mathematically expressed as given below,

$$\mathrm{RR} = \left[\frac{\mathrm{Tp}}{\mathrm{Tp+Fn}}\right] * 100 \qquad ...(8)$$

From (8), the recall rate ‘RR’ is calculated, Tp denotes the number of documents correctly retrieved, Fn indicates a number of relevant documents not retrieved. The recall rate is measured in terms of percentage (%).

The precision rate is defined as the ratio of relevant documents that are correctly retrieved to the total number of relevant and irrelevant documents retrieved. Therefore, the precision rate is expressed as given below,

$$\mathrm{P} = \left[\frac{\mathrm{Tp}}{\mathrm{Tp+Fp}}\right] * 100 \qquad ...(9)$$

From (9), the precision ‘P’ is calculated, Tp denotes a true positive i.e. number of documents correctly retrieved, Fp indicates a false positive i.e. number of irrelevant documents retrieved. The precision rate is measured in terms of percentage (%).

Computational complexity is defined as the amount of time taken by the algorithm to extract the dimensionality reduced features. Therefore, the overall complexity is measured as given below,

$$\mathrm{CC} = \mathrm{N} * \mathrm{Time\ (DRF)} \qquad ...(10)$$

Where, CC indicates a computational complexity, 'DRF' indicates a dimensionality reduced features, N indicates a number of documents. The computational complexity is measured in terms of milliseconds (ms).

4. Performance Analysis

In this section, performance analysis of the proposed NTFPR-BADC technique and existing methods improved the spherical k-means clustering algorithm [1], ADC [2] are evaluated in the form of a table and graphical representation. The statistical evaluation of the different methods are briefly described as given below,

Table 2. Comparison of Recall Rate

Number of documents	Recall rate (%)		
	NTFPR-BADC	**Improved spherical k-means clustering algorithm**	**ADC**
20	94.11	87.5	81.25
40	91.89	85.29	80.64
60	90	85.71	80.85
80	89.04	86.11	76.56
100	87.91	85.05	78.31
120	88.18	86.36	83.33
140	88	86.20	82.60
160	86.20	84.50	83.33
180	84.84	82.27	80.64
200	85.71	83.33	81.81

Table II describes the performance results of the recall rate with respect to the number of documents collected from the dataset. The number of documents is taken in the counts from 20 to 200. As shown in Table II, the various results of the recall rate are obtained for each clustering method. The observed results indicate that the NTFPR-BADC technique achieves a higher recall rate than the other existing clustering technique. Let us consider 20 documents for calculating the recall rate. The true positive and false negatives of the NTFPR-BADC technique are 16 and 1. Therefore, the recall rate is observed as 94.11%. By applying the similar counts of documents, the recall rate of the other two existing methods [1] [2] is 87.5% and 81.25%

respectively. Likewise, remaining runs are carried out with respect to different counts of documents. The overall observed results indicate that the NTFPR-BADC technique achieves a higher recall rate by 4% when compared to the Improved spherical k-means clustering algorithm [1] and 10% when compared to ADC [2].

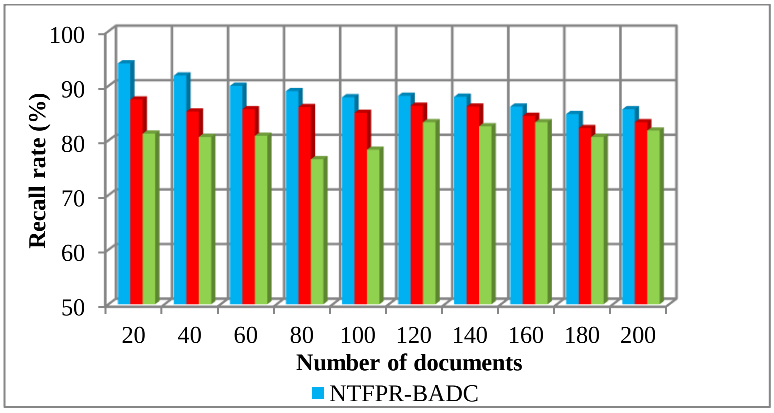

Fig. 4 performance results of recall rate

Figure 4 presents the performance results of the different clustering algorithms using the number of documents in the ranges from 20 to 200. As shown in figure 4, the expected improved performances are achieved using the NTFPR-BADC technique. However, the existing clustering algorithms show significantly lower performances. The NTFPR-BADC technique uses the bootstrap aggregative clustering technique partitions the documents into different clusters and it is stored into the cloud server. Then the cloud server displays the clustering results according to the user query. As a result, the relevant documents are correctly retrieved resulting in it increases the true positive.

Table 3. Comparison of Precision Rate

Number of documents	Precision rate (%)		
	NTFPR-BADC	**Improved spherical k-means clustering algorithm**	ADC
20	84.21	77.77	76.47

40	91.89	80.55	73.52
60	81.81	79.24	74.50
80	90.27	88.57	75.38
100	89.88	85.05	79.26
120	90.65	86.36	82.52
140	88	80.64	79.16
160	88.02	86.95	83.94
180	90.32	85.52	83.33
200	85.71	81.39	79.41

Table III presents the experimental assessment of the three techniques namely the NTFPR-BADC technique and existing methods improved spherical k-means clustering algorithm [1], ADC [2]. Among the three techniques, the performance of the precision rate using the NTFPR-BADC technique is higher than the existing methods. Let us consider the 20 documents, the precision rate of the proposed NTFPR-BADC technique is 84.21%. The precision rate of existing methods [1] [2] is 77.77% and 76.47% respectively. The results of the NTFPR-BADC technique are compared to existing methods. Ten runs were carried out and the average value proves that the precision rate is considerably increased by 6% and 12% when compared to existing methods. The overall graphical results are shown in figure 5.

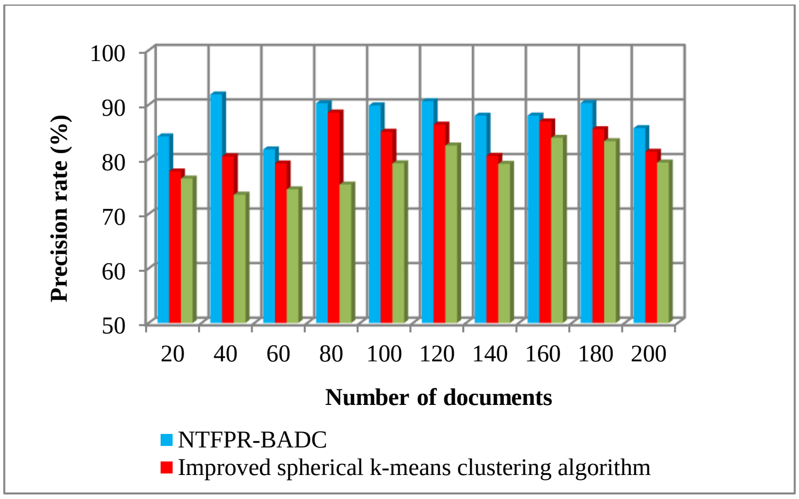

Fig. 5 performance results of the precision rate

Figure 5 illustrates the convergence behaviours of three different algorithms under datasets. As shown in the graphical illustration, the proposed document clustering technique outperforms well than the baselines approaches. The precision rate is measured based on the true positive and false positive. This significant improvement is achieved by the NTFPR-BADC technique using bootstrap aggregative document clustering. The ensemble technique uses the SFR clustering technique as a weak hypothesis. The clustering is performed based on the extracted features. The voting scheme is applied to discover the final ensemble clustering technique. The bootstrap aggregative clustering combines the weak hypothesis into a strong one. Therefore, the ensemble clustering technique groups the documents with higher accuracy. The strong clustering results are used for retrieving the exact document and minimizing the incorrect document retrieval.

Table 4 comparison of computational complexity

Number of documents	Computational complexity (ms)		
	NTFPR-BADC	Improved spherical k-means clustering algorithm	ADC
20	17	20	24
40	20	24	28
60	27	30	36
80	35.2	38.4	42.4
100	50	55	60
120	60	63.6	68.4
140	72.8	77	84
160	96	100.8	110.4
180	100.8	106.2	115.2
200	110	114	124

Table IV describes the experimental results of computational complexity with respect to different counts of documents taken in the range from 20 to 200. Compared to all the methods, the NTFPR-BADC technique consumes a lesser amount of time for selecting the dimensionality reduced features. Let us consider 20 documents, the time is taken to find the dimensionality reduced features of the NTFPR-BADC technique was found to be '17ms'. However, the time consumed for detecting the features using an Improved spherical k-means clustering algorithm [1], ADC [2] was found to be '20ms' and 24ms.

From the statistical analysis, it is inferred that the NTFPR-BADC technique minimizes the computational complexity. After obtaining the ten results, the observed computational complexity is compared to other baseline clustering techniques. The average value shows that the proposed clustering technique reduces the computational complexity by 8% and 19% as compared to the improved spherical k-means clustering algorithm [1], ADC [2] respectively.

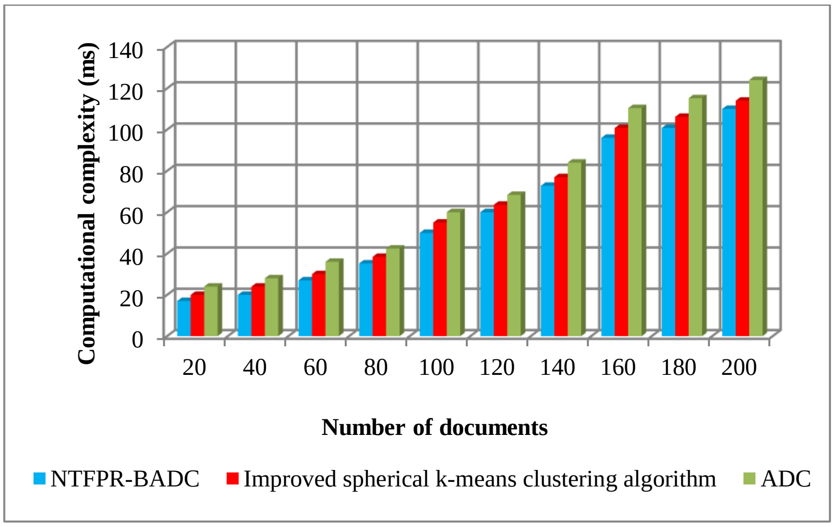

Fig. 6 performance results of computational complexity

Figure 6 illustrates the convergence graph of the computational complexity using three different clustering techniques. The above graph shows that the computational complexity is increased while increasing the number of documents. As shown in the graph, the different colors of the columns indicate the computational complexity of three different clustering techniques. The blue color column shows the computational complexity of the NTFPR-BADC technique and computational complexity of improved spherical k-means clustering algorithm [1], ADC [2] are indicated by the red and green color respectively. Among three clustering techniques, the NTFPR-BADC technique outperforms well in terms of achieving lesser complexity.

This is due to the application of the target feature projective regression for analyzing the words based on the term frequency measure from the given input documents.

5. Conclusion

In this paper, a novel clustering technique NTFPR-BADC is employed for considerably improving the performance of document clustering in the cloud. The pre processing step of the NTFPR-BADC technique obtains the major words and eliminates the other words to minimize the time consumption of document categorization. The target projective pursuit regression is applied in NTFPR-BADC for analyzing the words based on the word frequency measure to extract the features from the documents to minimize the computational complexity. The final step is to partition the given input documents based on the extracted keywords using the bootstrap aggregative clustering technique. The clustered documents are stored in the cloud for accurate retrieval with minimum time. We evaluate the NTFPR-BADC technique through in-depth experiments with the document dataset and compare the results with two different clustering baselines algorithms. Results have proved that the proposed NTFPR-BADC technique has enhanced the performance of bootstrap aggregative clustering in terms of standard evaluation measurement criteria and the average computational complexity of the algorithm is also reduced. Besides, the statistical analysis of precision and recall rate is higher using the NTFPR-BADC technique than the conventional clustering techniques.

References

1. Hyunjoong Kim, Han Kyul Kim, Sungzoon Cho, “Improving spherical k-means for document clustering: Fast initialization, sparse centroid projection, and efficient cluster labeling”, Expert Systems with Applications, Elsevier, Volume 150, 2020, Pages 1-12
2. Jinuk Park, Chanhee Park, Jeongwoo Kim, Minsoo Cho, Sanghyun Park, “ADC: Advanced document clustering using contextualized representations”, Expert Systems with Applications, Elsevier, Volume 137, 2019, Pages 157-166

3. Laith Mohammad Abualigah, Ahamad Tajudin Khader, Essam Said Hanandeh, “A combination of objective functions and hybrid Krill herd algorithm for text document clustering analysis”, Engineering Applications of Artificial Intelligence, Volume 73, 2018, Pages 111–125
4. Ahmad Muqeem Sheri, Muhammad Aasim Rafique, Malik Tahir Hassan, Khurum Nazir Junejo, and Moongu Jeon “Boosting Discrimination Information Based Document Clustering Using Consensus and Classification” IEEE Access, Volume 7, 2019, Pages 78954 – 78962
5. Xiaobing Pei, Chuanbo Chen, Weihua Gong, “Concept Factorization With Adaptive Neighbors for Document Clustering”, IEEE Transactions on Neural Networks and Learning Systems, Volume 29, Issue 2, 2018, Pages 343 – 352
6. Amir Karami, “Application of Fuzzy Clustering for Text Data Dimensionality Reduction”, International Journal of Knowledge Engineering and Data Mining, Volume 6, Issue 3, 2019, Pages 1-19
7. Hossam M. J. Mustafa, Masri Ayob, Dheeb Albashish, Sawsan Abu-Taleb, “Solving text clustering problem using a memetic differential evolution algorithm”, PLoS ONE, Volume 15, Issue 6, 2020, Pages 1-18
8. Laith Mohammad, Abualigah Ahamad, Tajudin Khader, Essam Said Hanandeh, “A new feature selection method to improve the document clustering using particle swarm optimization algorithm”, Journal of Computational Science, Elsevier, Volume 25, 2018, Pages 456-466
9. Junkai Yi, Yacong Zhang, Xianghui Zhao, and Jing Wan, “A Novel Text Clustering Approach Using Deep-Learning Vocabulary Network”, Mathematical Problems in Engineering, Hindawi, Volume 2017, March 2017, Pages 1-13
10. Karthick Seshadri, K. Viswanathan Iyer, Mercy Shalinie, “Design and evaluation of a parallel document clustering algorithm based on hierarchical latent semantic analysis”, Concurrency and Computation Practice Experience, Wiley, Volume 31, Issue 13, 2019, Pages 1-20
11. Ruksana Akter and Yoojin Chung, “An Improved Genetic Algorithm for Document Clustering on the Cloud”, International Journal of Cloud Applications and Computing, Volume 8, Issue 4, 2018, Pages 20-28
12. Neha Agarwal, Geeta Sikka, Lalit Kumar Awasthi, “Enhancing web service clustering using Length Feature Weight Method for service

description document vector space representation", Expert Systems with Applications, Elsevier, Volume 161, 2020, Pages 1-11

13. Peng Yang, Yu Yao, Huajian Zhou, "Leveraging Global and Local Topic Popularities for LDA-Based Document Clustering", IEEE Access, Volume 8, 2020, Pages 24734 – 24745
14. Ruizhang Huang, Weijia Xu, Yongbin Qin, Yanping Chen, "Hierarchical Dirichlet Multinomial Allocation Model for Multi-Source Document Clustering", IEEE Access, Volume 8, 2020, Pages 109917 – 109927
15. R. Janani and S. Vijayarani, "Text document clustering using Spectral Clustering algorithm with Particle Swarm Optimization", Expert Systems with Applications, Elsevier, Volume 134 2019, Pages 192-200
16. Jian-Ping Mei, "Semi supervised Fuzzy Clustering With Partition Information of Subsets", IEEE Transactions on Fuzzy Systems, Volume 27, Issue 9, 2019, Pages 1726 – 1737
17. Maria Th. Kotouza, Fotis E. Psomopoulos and Pericles A. Mitkas, "A dockerized framework for hierarchical frequency-based document clustering on cloud computing infrastructures", Journal of Cloud Computing, Springer, Volume 9, 2020, Pages 1-17
18. Rasim M. Alguliyev, Ramiz M. Aliguliyev, Nijat R. Isazade, Asad Abdi, Norisma Idris, "COSUM: Text summarization based on clustering and optimization", Expert Systems, Wiley, Volume 36, Issue 1, 2019, Pages 1-17
19. Juan Zamora, Héctor Allende-Cid, Marcelo Mendoza, "Distributed Clustering of Text Collections", IEEE Access, Volume 7, 2019, Pages 155671 – 155685
20. Rui Zhao, Kezhi Mao, "Fuzzy Bag-of-Words Model for Document Representation", IEEE Transactions on Fuzzy Systems, Volume 26, Issue 2, 2018, Pages 794 - 804