academicJournals

Vol. 10(1), pp. 14-22, 15 January, 2015
DOI: 10.5897/SRE2014.6107
Article Number:549B62F49534
ISSN 1992-2248
Copyright©2015
Author(s) retain the copyright of this article
http://www.academicjournals.org/SRE

Scientific Research and Essays

Full Length Research Paper

An efficient hybrid distributed document clustering algorithm

J. E. Judith[1]* and J. Jayakumari[2]

[1]Department of CSE, Noorul Islam Centre for Higher Education, Kumaracoil, India.
[2]Department of ECE, Noorul Islam Centre for Higher Education, Kumaracoil, India.

Received 3 September, 2014; Accepted 23 December, 2014

Recent advances in information technology have led to an increase in volumes of data thereby exceeding beyond petabytes. Clustering distributed document sets from a central location is difficult due to the massive demand of computational resources. So there is a need for distributed document clustering algorithms to cluster documents using distributed resources. The greatest challenge in this area of distributed document clustering is the clustering quality and speedup associated with increase in document sets. The proposed clustering algorithm uses a hybrid algorithm which comprises of Particle Swarm Optimization (PSO), K-Means clustering and Latent Semantic Indexing (LSI) algorithm (PKMeansLSI), and uses MapReduce framework for distributed computation. The resultant of this is that it ultimately promotes clustering quality of the algorithm. The MapReduce framework and its corresponding implementation Hadoop is used as a distributed programming model which stresses on the improvement factor of the speedup of algorithm. The execution time is dramatically reduced as the dimensionality of documents is reduced. Experiment results show improved quality and effectiveness of the hybrid algorithm with varying increase in document size.

Key words: Distributed document clustering, Hadoop, K-Means, particle swarm optimization (PSO), latent semantic indexing (LSI), MapReduce.

INTRODUCTION

Data mining is the process that attempts to discover patterns in large data sets. Distributed Data Mining (DDM) (Datta et al., 2009) is one of the important and active areas of research due to the challenges and applications associated with the problem of extracting previously unknown knowledge from very large real-world databases. Document clustering groups similar documents into a single cluster. To cluster documents accurately the similarity between a pair of documents must be defined (Anna Huang, 2008). The quality of information retrieval in both centralized and decentralized environments can be improved by using an advanced clustering framework (Khaled et al., 2009). Distributed document clustering algorithms perform clustering based on the availability of the distributed resources (Eshref et al., 2003). Along with the recent advances in algorithmic and conceptual changes an advanced clustering framework is needed for processing large amount of distributed document datasets. The reason is due to the decentralization of huge volume of documents to be

*Corresponding author. E-mail: judithjegan@gmail.com processed (Datta et al., 2009) and the inability of the large amount of documents to be processed by central supercomputers. Centralized data warehouse based mining cannot scale (Khaled et al., 2009) to that extend. Storage and processing of mass documents in a distributed environment can solve this problem. It is difficult to handle problems like data distribution, fault tolerance and system communication that occur in such parallel and distributed environment. In order to solve these problems, new tools, technologies and frameworks for distributed processing (Surendra and Xian-He, 2011) are emerging. One of the most popular of these emerging technologies is Hadoop, an open source software framework for handling large amount of distributed data in distributed environment.

MATERIALS AND METHODS

Distributed document clustering

Recently different distributed document clustering algorithms have been proposed to cluster documents from distributed resources. The main objective of these algorithms is to move the computation (clustering algorithm) to the documents in each node of distributed site instead of moving all the documents to a central node and then performing the computation. A local model is computed by applying clustering algorithm to each node and is aggregated to produce optimized clusters. The issues in the distributed clustering algorithms can be categorized in to algorithmic issues and implementation issues. Recently many conceptual and algorithmic changes (Yang et al., 2006; Datta et al., 2009; Khaled et al., 2009; Odysseas et al., 2011; Hu et al., 2013) have been made to these traditional clustering algorithms by adding many concepts like fuzzy theory, swarm intelligence, genetic algorithms, ontology, wordnet, word sense disambiguation and many more to increase the efficiency and quality of the algorithm. The implementation issues are related to the distributed environment in which the distributed document clustering algorithms are studied. An exhaustive review on recent research of distributed document clustering algorithms for distributed environments like peer-to-peer networks is emphasized with top concerns to the clustering quality.

Datta et al. (2009) proposed an approximate P2P K-means algorithm which requires that each node must synchronize only with the nodes that is connected directly to it. This is more effective for a dynamic network but still it is sensitive to the distribution of documents to each peer. Quality of clusters generated is of concern. Hammouda and Kamel (2009) proposed a hierarchically distributed peer-to-peer clustering algorithm (HP2PC) for large P2P systems. In this work K-means algorithm is applied to the data in each peer node to generate a set of centroids that are passed to the neighbors until it reaches the supernode which contains the centroids of the whole dataset. Scalability is of concern as more and more nodes are added to the network. The authors determined the clustering quality based on the skewness of similarity histograms of the individual cluster which deteriorates with increase in hierarchy. Decentralized Probabilistic Text Clustering for peer-peer networks is proposed by Papapetrou et al. (2012). This work uses a probabilistic approach using Distributed Hash Table (DHT) for assigning documents to clusters which increases the scalability algorithm but there is a decrease in speed-up with increase in dataset. Clustering quality is of concern in this work. Thangamani and Thangaraj, (2012) proposed an effective fuzzy semantic clustering scheme for decentralized network through multi domain ontology model. The results show better clustering results using fuzzy concept with semantic concepts like ontology but still there are scalability issues. Thus the issues identified by the clustering algorithm in the peer-to-peer environment are scalability, speedup, and distribution of input data. This can be overcome by making possible tweaks in implementation of the algorithm on distributed environment where it is studied. Also this traditional distributed computing approach might not be able to meet the next generation requirement of distributed processing. Many easy-to-use distributed processing tools have evolved to handle the drastic increase in data. Hadoop MapReduce framework (Wan et al., 2009; Lei Qin et al., 2011; Ping et al., 2011) can be used for distributed computation which overcomes these issues to improve the algorithmic performance. The proposed work overcomes the implementation issues identified in peer-to-peer environment using a scalable tool for distributed processing called Hadoop. A review was meticulously carried out to mine knowledgeable data using Hadoop.

MapReduce based distributed Latent Semantic Indexing (LSI) and K-means for document clustering is proposed by Yang et al. (2010). It provides comparison with standalone LSI and distributed k-means LSI using socket programming. The result shows great improvement in speedup and scalability. A distributed MST (minimum spanning tree) algorithm based on MapReduce programming model was proposed by Kehua et al. (2012). A distributed MST text clustering algorithm is designed and implemented and its performance is compared. The speed-up of the algorithm can still be improved. MapReduce based particle swarm optimization clustering (MR-CPSO) algorithms is proposed by Ibrahim et al. (2012). Clustering is considered as optimization problem which is used to find the best solution. The results show that the scalability of MR-CPSO is high when there is an increase in dataset size. Liu and Ge (2012) proposed a MapReduce based name disambiguation system. The document clustering task is parallelized by dividing data to a number of maps and reduces and disambiguation is performed using LSI. Hu et al. (2013) proposed a Fuzzy Approach to Cluster Text Documents Based on MapReduce. Fuzzy set is used to categorize text documents. A parallel text clustered framework is designed based on MapReduce according to the proposed text clustering procedure. Patil and Nandedkar, (2014) proposed a MapReduce based K-Means and hierarchical clustering algorithm which shows improvement in performance but lacking clustering quality.

To the best of the authors none of the algorithms consider the hybrid of MapReduce based PSO-KMeans-LSI that improves the quality and performance of clustering. The proposed work aims in improving the speed-up and quality of clustering algorithm.

Particle swarm optimization algorithm

Clustering is considered as optimization problem using PSO. It is used to find optimal cluster centroid rather than finding optimal partition. These optimal centroids are found for minimizing the intra-cluster (within) distance as well as maximizing distance between clusters. PSO performs globalized searching (Ibrahim et al., 2012) in order to determine optimal centroids. PSO algorithm is based on social behavior of birds flocking. Birds in a flock are represented as particles. Each particle is considered as a document. A particle contains information like location and velocity. A particles location represents one solution. A new solution is generated when the particle moves to a new location. This new solution is evaluated using fitness function in Equation (1), which is the average distance between document and cluster centroids.

$$f = \frac{\sum_{i=1}^{N_C} \left\{ \frac{\sum_{j=1}^{P_i} d\left(t_i, n_{ij}\right)}{P_i} \right\}}{N_c} \tag{1}$$

Where *d (t_i, n_j)* is the distance between document n_{ij} and the cluster centroid $t_{i.,}$ P_i is the document number, N_c is the cluster number. The velocity and position of new particle are updated based on the following equations:

$$v_{id} = w * v_{id} + c_1 * rand_1 * (p_{id} - xi_d) + c_2 * rand_2 * (p_{gd} - x_{id}) \tag{2}$$

$$x_{id} = x_{id} + v_{id} \tag{3}$$

This process is repeated for maximum number of iterations. The optimal centroids are generated using this method.

K-Means clustering algorithm

K-Means algorithm is sensitive to the selection of initial cluster centroids and uses these centroids for maximizing intra-cluster similarity (within) and minimizing inter-cluster similarity. It performs localized searching to determine the initial centroids (Wan et al., 2009). K-Means clustering uses randomly generated seeds as initial cluster centroids. Each document is compared to all the cluster centroids (Datta et al., 2009). The document is assigned to the cluster based on the similarity (Anna, 2008). The Jaccard similarity measure used is described as:

$$SIM_J\left(\vec{t}_a, \vec{t}_b\right) = \frac{\vec{t}_a . \vec{t}_b}{\vec{t}_a^{\,2} + \vec{t}_b^{\,2} - \vec{t}_a . \vec{t}_b} \tag{4}$$

Where t_a and t_b are n-dimensional vectors over the term set. It compares the sum weight of terms shared to the sum weight of terms in any of the two documents but is not the terms shared. The cluster centroids are recalculated as the mean of the document vectors that belong to that cluster using the following Equation (5):

$$C_j = \frac{1}{n_j} \sum_{d_j \in Q_j} d_j \tag{5}$$

Where n_j is the number of document vectors that belong to cluster Q_j and d_j is the document vector that belong to Q_j.

Latent semantic indexing (LSI) algorithm

LSI is a method of dimensionality reduction that can improve the efficiency of clustering. It analyzes the whole document-term matrix (Jianxiong and Watada, 2011; Yang Liu and Ge, 2010), and projects it in a lower dimensional latent space. LSI has the ability to correlate semantically related terms by establishing associations between terms. Latent semantic indexing applies a linear algebra technique, called Singular Value Decomposition (SVD), to a document- term matrix. It generates a document-term matrix which represents this original document-term matrix approximately. This matrix not only reduces the scale of the original matrix but also shows relationship among terms. SVD decomposes document-term matrix $A_{t\times d}$ to a product of three matrices: $T_{t\times n,} S_{n\times n,} D_{d\times n}$: $A_{t\times d} = T_{t\times n} S_{n\times n} \left(D_{d\times n}\right)^T$. These matrices are then reduced to the given number of dimensions *k* such that *k<<n* to result in truncated matrices $T_{t\times k}, S_{k\times k}, \left(D_{d\times k}\right)^T$. These matrices are multiplied to give a new matrix $A_{t\times k} = T_{t\times k} S_{k\times k} \left(D_{d\times k}\right)^T$ which is the least square best fit approximation of matrix *A* with k singular values. The given number of dimensions is the *k* singular values. Using SVD on PSO-KMeans (PKMeans) clusters enhances the performance by capturing the important semantic structure in the association of terms and therefore reducing the dimensionality.

OVERVIEW OF THE PROPOSED METHODOLOGY

The different steps followed in this proposed methodology are summarized as:

1. Choosing a corpus of documents.
2. Preprocessing the text documents and Vector Space Model representation based on MapReduce.
3. MapReduce based K-Means clustering using PSO generated optimal centroids (PKMeans).
4. Dimensionality reduction using LSI on PKMeans clusters (PKMeansLSI).
5. Generation Optimized dimensionality reduced document clusters.

Document preprocessing and representation based on MapReduce

Document preprocessing and vector space representation is done using MapReduce framework for efficient representation of the documents. It takes a set of input plain text document and transforms it in to a form (Datta et al., 2009) to be included in the vector space model. These preprocessing steps are performed in parallel using MapReduce programming methodology. Some common words like stopwords are removed. Stemming is done to reduce words to their base form or stem. Porter's algorithm (Porter, 1980) is the defacto standard used for stemming. In order to represent the documents using Vector Space Model (Salton et al., 1975), documents have to be transformed from full text version to document vector which describes the content of the document as a vector. Each document is represented by a vector *d = tf₁, tf₂… tfₙ*, where tf_i is the frequency of each term (TF) in the document. The tf_i*idf_i representation of the documents is done in parallel using MapReduce methodology.

In order to represent the documents in the same term space, the number of times term appears in a given document, number of terms in each document, number of documents in which the given term appears and the total number of documents are determined. Thus, each component of the vector d now becomes tf_i*idf_i. This is represented on a document-term matrix (Jianxiong and Watada, 2011). This represents the term weight of the document. The frequency of a term t in the document d gives the term weight of the document d in a collection of documents D that is described as:

$$tfidf\left(d,t\right) = tf\left(d,t\right) \times \log\left(\frac{|D|}{df(t)}\right) \tag{6}$$

Where *df(t)* is the frequency of documents in which term t appears and *tf (d,t)* is the frequency of term *t* in document *d.*

Proposed distributed document clustering algorithm based on MapReduce

The proposed algorithm is based on MapReduce methodology on Hadoop framework. It provides the ability to transparently distribute the documents (Lei et al., 2011; Ping et al., 2011) to one or more

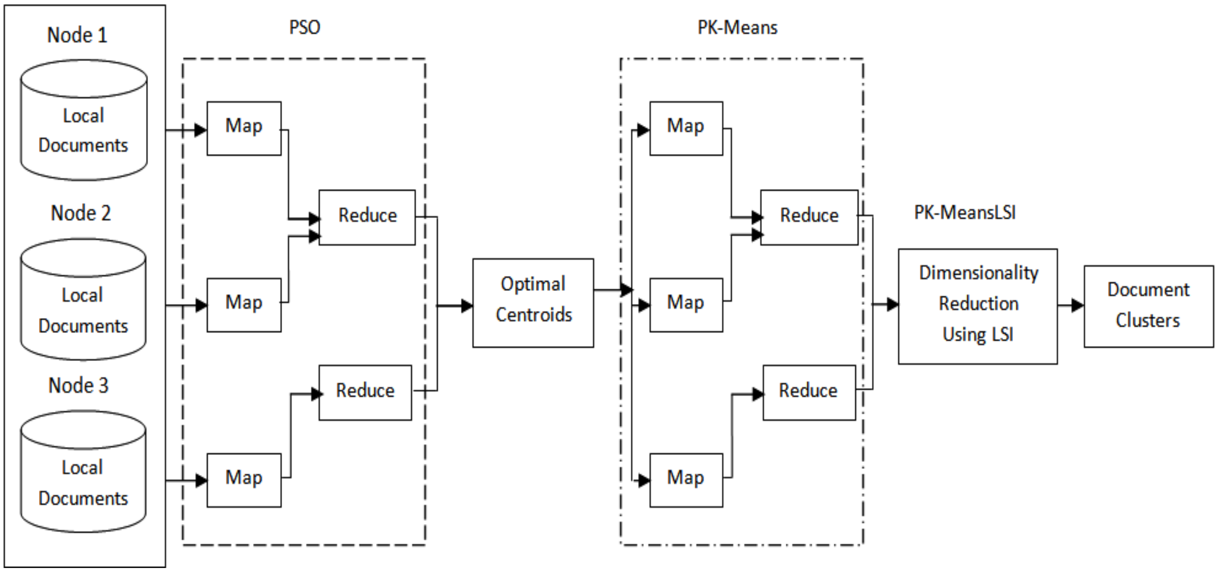

Figure 1. Proposed Distributed Clustering Algorithm based on MapReduce Framework**.**

storage entities and apply operations to each subsets using Hadoop. The proposed algorithm consists of two phases of MapReduce operations. The Phase I of MapReduce operation is for the generation of optimal centroids using PSO, whereas Phase II of MapReduce is for the purpose of KMeans clustering using PSO generated centroids. Figure 1 depicts the complete methodology of the proposed distributed document clustering algorithm. Latent Semantic Indexing (LSI) technique is applied to the resultant document-term matrices which truncates the matrices to reduced dimensions and describes the relationship between terms. The proposed algorithmic steps are given as follows: Hadoop Distributed File System (HDFS) stores the input document vectors and the initial input document centroids.

Phase I

1. The Map function splits the input documents into several data blocks (64 MB each) with the initial document velocity and position. The fitness evaluation function evaluates the position of document vectors and is assigned a fitness value. Fitness function is evaluated as the average distance between the document and cluster centroids as in Equation (1). The document position with the highest fitness value in the entire document set is considered the global best solution.
2. The document velocity and position values are updated based on the Equation (2) and Equation (3).
3. Repeat the steps 1 to 3 until maximum number of iteration is reached. The number of iteration is fixed to 100.
4. The reduce function updates document vector based on the new centroids. This process is repeated until all the document vectors are updated.
5. The optimal centroids generated are stored in HDFS along with the input document vectors.

Phase II

1. The Map function splits the input documents in to several data blocks (64 MB each). The similarity between the input document vectors and PSO generated optimal centroids are evaluated for each data block using Jaccard similarity given in Equation (4).
2. The reduce function collects the map outputs and updates the centroids as the mean of all cluster documents.
3. All the optimal centroids are aggregated and the dimensionalities of centroids generated are reduced based on Latent Semantic Indexing technique. This performs singular value decomposition on the resultant smaller matrices. This reduces the overhead in computation which increases the speed of the algorithm.

EXPERIMENTAL RESULTS

The distributed environment is set up using Hadoop cluster environment. Each node of the cluster consists of Intel i7 CPU 3GHz, 1TB of local hard disk storage reserved for HDFS and 8 GB for main memory. All nodes are connected by standard gigabit Ethernet network on a flat network topology. Parallel jobs are submitted on the parallel environment like Hadoop (MapReduce). The Reuters document dataset (Reuters-21578 text collection distribution 1.0) and several other massive sizes of document datasets RV1 was taken up for extensive processing across distributed nodes, which of these labeled a totaling of approximately 36398MB in size, comprising of about 860 diverse subjects from almost every domains of the knowledge base.
There are a variety of evaluation metrics in order to evaluate the performance (Anna, 2008; Datta et al., 2009; Khaled et al., 2009) of the proposed clustering algorithm. The performance is evaluated by varying the document size and the number of nodes. The metric used for evaluation is clustering quality and execution time. The performance metric used to evaluate quality of the clustering is purity. The execution time and speedup of the proposed algorithm is also evaluated.

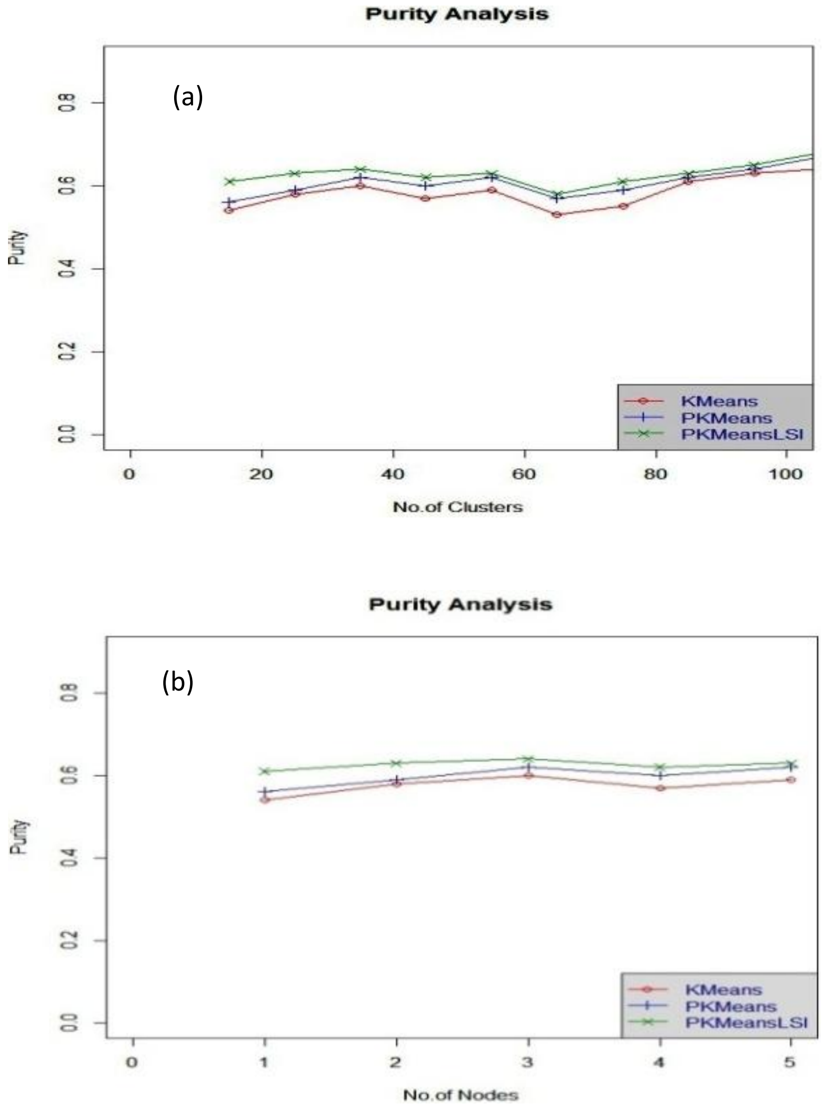

Figure 2. Purity Analysis by varying the number of clusters and number of nodes.

Purity

This metrics is used to evaluate whether the documents in a cluster are from a single category (Anna, 2008). Purity of C_j is formally defined as:

$$P\left(C_j\right) = \frac{1}{n_j} max_h\left(n_j^h\right) \tag{7}$$

Where $max_h\left(n_j^h\right)$ is the documents that are from the main category in cluster C_j and n_j^h represents the number of documents from cluster C_j assigned to category h. For an ideal cluster the purity value is 1 because it contains documents.

The PSO parameters used for evaluation are inertia weight w = 0.72 and the acceleration constants c1 and c2 are set to 1.7. Figure 2a and b shows that the purity values of the proposed algorithm increase while varying the number of clusters and nodes. The results show that the proposed hybrid algorithm is able to assign documents to the correct cluster with increased purity value of 0.75 when compared to standalone KMeans algorithm. It was observed that the purity value of clustering results after 50 iterations are better than the K-Means clustering results.

Execution time

The execution time of the proposed hybrid algorithm is evaluated by measuring the time taken for clustering different document sizes and by increasing the number of nodes.

Figure 3 shows the execution time analysis of the

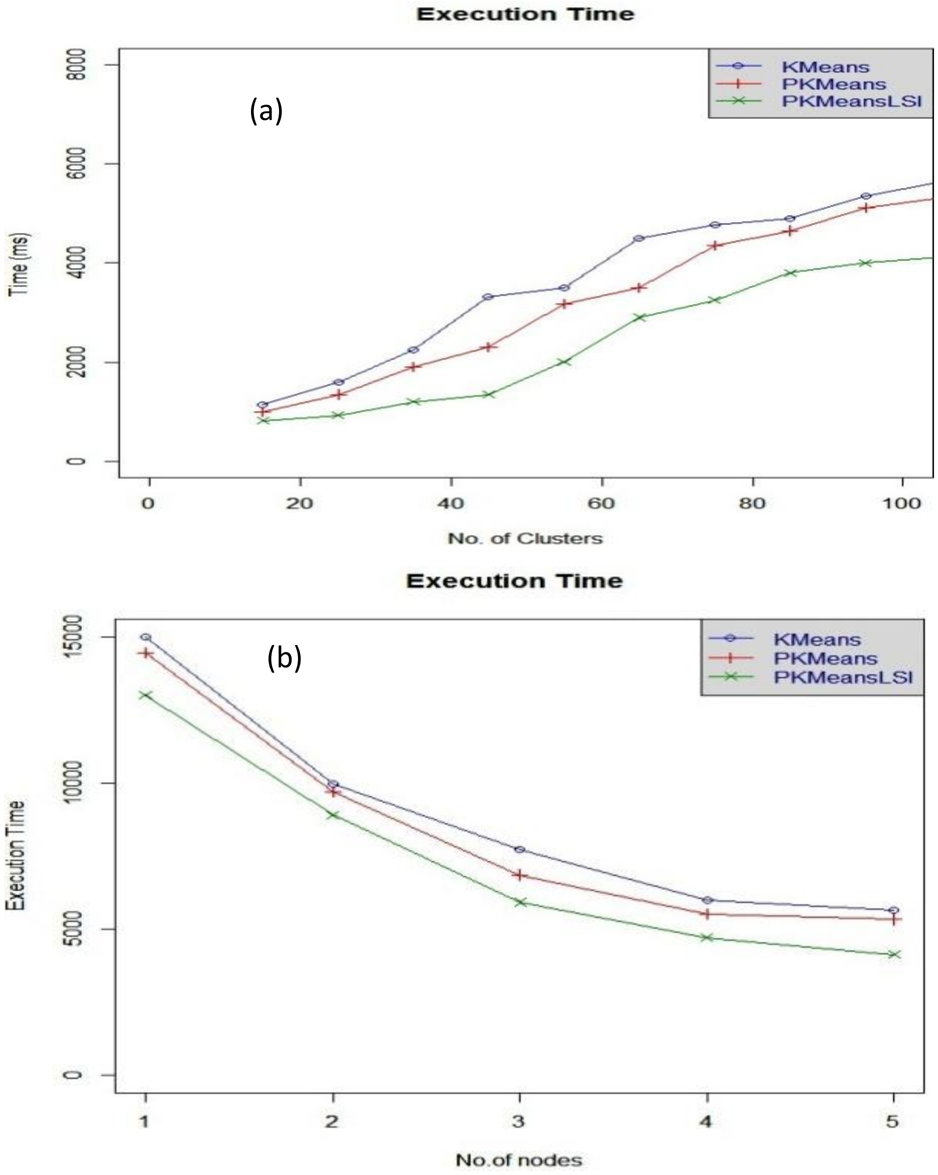

Figure 3. Execution time analysis by varying the number of clusters and number of nodes. (a) Execution time analysis based on number of clusters; (b) Execution based on number of nodes

proposed algorithm when the number of clusters and nodes are increased. When the number of clusters is increased, the execution time increases. Figure 3a shows that the execution time taken by the proposed algorithm is less when compared to time taken by MapReduce based K-Means algorithm. Figure 3b shows that the execution time decreases almost linearly with increasing number of nodes of the Hadoop cluster. The execution time in a single node system is high and is decreased when the number of nodes increases.

Speedup

Speedup is the relative increase in speed of one algorithm over the other. In this paper it is measured as the relative increase in speed of the proposed algorithm on a standalone machine that uses Local File System (LFS) to the relative increase in speed of the proposed algorithm on a Hadoop cluster that uses Hadoop Distributed File System (HDFS). The speedup of the proposed algorithm is determined as:

$$Speedup \;\; S_p = T_c / T_d$$
$$S_p = T_{LFS} / T_{HDFS} \tag{8}$$

Figure 4 depicts that the speedup of the proposed algorithm increases linearly but is stable with increase in the number of clusters. Figure 5 describes the

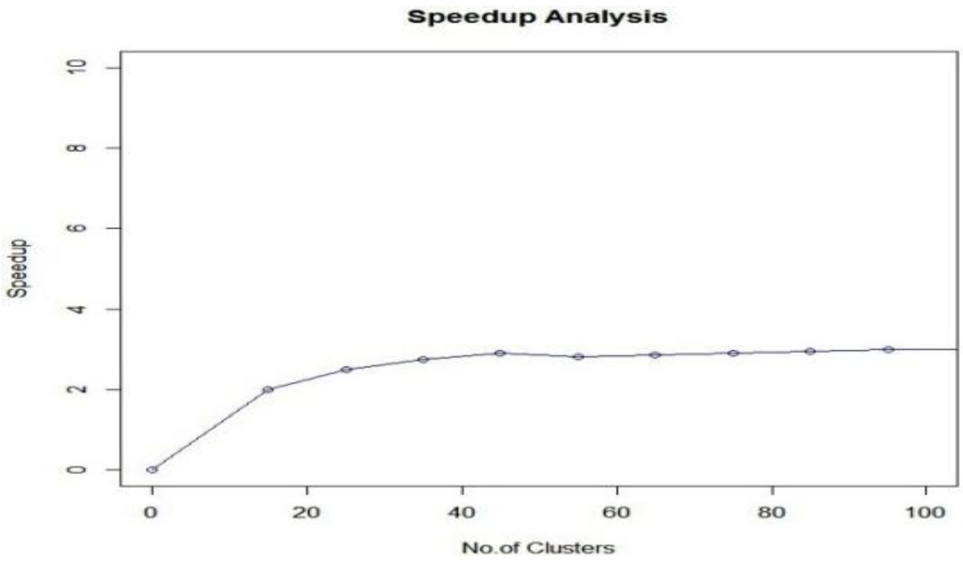

Figure 4. Speedup analysis.

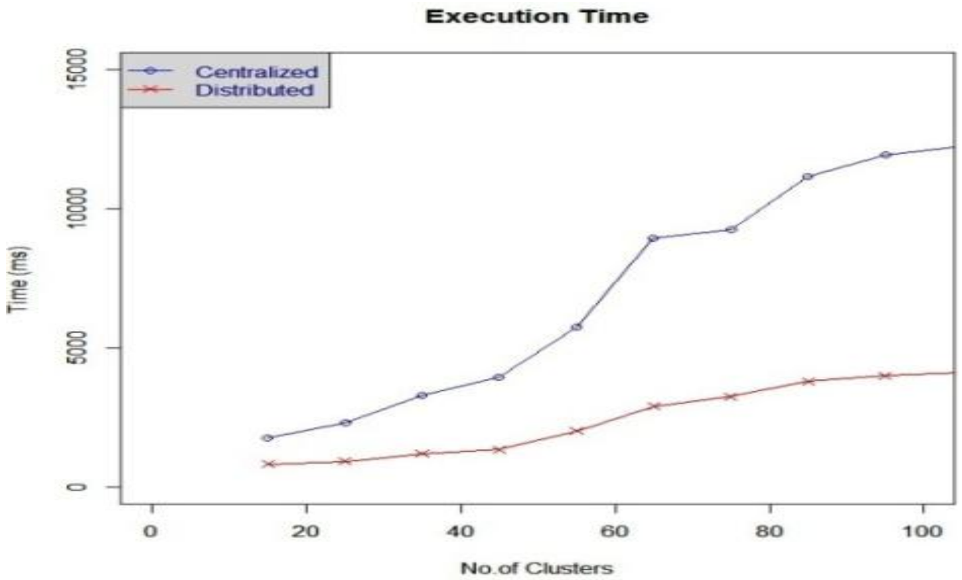

Figure 5. Execution time comparison of the proposed algorithm.

performance of the proposed algorithm in a standalone system that uses LFS (Local File System) to the performance of the proposed algorithm that uses Hadoop cluster.

DISCUSSION

Extensive experiments were performed on the proposed MapReduce based hybrid clustering algorithm which addresses performance issues such as increasing the quality of clusters and reducing the execution time. The impact of PSO algorithm on the performance of K-means clustering algorithm is that it ultimately performs a globalized search to find the best solution for the clustering process. This dramatically overcomes the two major drawbacks of K-means (Cui et al., 2005) algorithm such as sensitivity to the selection of initial cluster centroids and the local optima convergence problem. PSO is an iterative algorithm that meticulously finds the optimal solution based on a specific similarity measure. This optimal solution is determined iteratively by using similarity measure called Jaccard coefficient.

Jaccard coefficient compares the sum weight of terms

Table 1. Purity analysis of algorithms.

No.	Algorithms	Purity values
1	K-Means	0.584
2	PK-Means	0.608
3	PK-MeansLSI	0.657

Table 2. Initial feature space dimensions using terms.

Data	Documents	Classes	Terms
Reuters 21578	21578	131	42686
RV1	1,000,000	860	3268650

present in the documents and its value usually ranges from 0 to 1. Apparently, the quality of cluster generated depends primarily on this similarity measure. In order to increase the quality of cluster, the intra-cluster similarity has to be maximized and inter-cluster similarity has to be minimized; from which we understand that the similarity of documents within cluster should be increased and the similarity of documents between clusters should be decreased. This membership of documents within a cluster depends on this similarity measure. The advantage of Jaccard coefficient measure is that it serves to find out more coherent clusters (Anna, 2008). In this paper the performance of K-means algorithm is improved using a PSO optimized centroids and the cluster membership is determined based on Jaccard similarity. The new centroids are recomputed after each iteration and all documents are reassigned based on these new centroids.

The impact of clustering algorithms on clustering quality is evaluated after each iteration. Purity is used as the quality measure to evaluate performance of the proposed clustering algorithm. Higher the purity values, better the clustering solutions (Anna, 2008). Table 1 shows the average relative purity values of the MapReduce based individual algorithms compared to the proposed hybrid algorithm for the given input document datasets. It describes that purity values of the proposed hybrid algorithm is improved when compared to the performance of standalone algorithms.

Latent Semantic Indexing (LSI) algorithm also has an impact on the purity values of the PSO optimized resultant clusters. Generally, the evaluation of LSI algorithm with changing parameter setting depends on the application targeted. LSI algorithm is can be used to reflect the semantic structure of documents. It can also be used as dimensionality reduction technique. Here, the performance of LSI is mainly targeted to improve the quality of clusters. Also the computational complexity of Singular Value Decomposition (SVD) involved in LSI is drastically reduced since the algorithm is applied to the resultant smaller document-term matrices of PKMeans algorithm. Table 2 describes the initial feature space dimensions of the input document data set.

In this paper the term frequency and document frequency TFIDF are combined as the feature weighting scheme, which is based on the idea that if a feature appears many times in a document, that feature must have more weight. A feature that appears in many documents is not important since it is not very useful to distinguish different documents. Therefore, it should have a lower weight. The impact of clustering quality on dimensions is determined using purity values. Dimensions are reduced using LSI based on *k* singular values. The original matrix of each cluster generated by K-Means algorithm is reduced to the k number of dimensions determined by the singular values of the original document term matrix. For experimental evaluation, the dimensions were increased from 50, 100, 150, 200, 250, and 300.

The algorithm is repeated for at least about 50 iterations. Results show that the purity values increase for lower dimensions and degrade as the dimensions increases. Table 3 shows the purity values for the document set under different dimensions. Also, it indicates that for a given range of dimensionality from 50 to 100, the purity value is high for the proposed hybrid algorithm and degrades with increase in dimensions.

The execution time of the proposed MapReduce based hybrid algorithm is statistically compared with the centralized hybrid algorithm as in Figure 5. In this paper Hadoop Distributed File System (HDFS) is primarily used for distributed storage while executing MapReduce algorithms. This enables automatic data distribution and assists in distributed storage of intermediate and final results. Eventually, it overcomes communication overhead and makes parallel execution of task a lot faster than ever. Basically, the centralized hybrid algorithm is based on Local File System (LFS). The impact of execution time on the performance of the proposed clustering algorithm is evaluated by comparing the execution time of HDFS based algorithm to LFS based algorithm. Figure 5 shows that the execution time consumed by Hadoop based hybrid algorithm is reduced when compared to centralized LFS based algorithm.

Thus a scalable hybrid PKMeansLSI algorithm is proposed using MapReduce distributed methodology to overcome the inefficiency of clustering for large datasets. It's indicated that the hybrid PKMeansLSI algorithm can be successfully parallelized with the MapReduce methodology running on commodity hardware. Most centralized clustering algorithms suffer from scalability problem with increase in dataset size, and are computationally expensive. Due to these aforementioned reasons, the distributed computation of data clustering algorithm is paramount in order to deal with large scale data. In order to develop a good distributed clustering algorithm that takes big data into consideration, it is

Table 3. Impact of clustering quality on dimensions for clustering algorithms.

MapReduce-KMeans		MapReduce-PKMeans		MapReduce-PKMeansLSI	
Dimensions	**Purity**	**Dimensions**	**Purity**	**Dimensions**	**Purity**
50	0.64	50	0.68	50	0.75
100	0.61	100	0.63	100	0.67
150	0.58	150	0.59	150	0.64
200	0.56	200	0.56	200	0.61
250	0.53	250	0.54	250	0.59
300	0.51	300	0.52	300	0.57

mandatory for an algorithm to be efficient, scalable and must generate high quality clusters.

Conclusion

In this paper an efficient hybrid distributed document clustering algorithm based on MapReduce methodology is proposed. This algorithm is based on K-Means algorithm that uses PSO generated centroids which shows improvement in quality as the purity values are increased .The execution time of the proposed clustering algorithm is decreased since the dimensionality is reduced based on LSI (Latent Semantic Indexing) technique. Hadoop MapReduce frame work is used for distributed processing which shows improvement in speedup and clustering quality. This research work finds application in areas of document summarization, digital libraries, information retrieval, scientific analysis etc.

Conflict of Interest

The authors have not declared any conflict of interest.

REFERENCES

Anna H (2008). Similarity Measures for Text Document Clustering. Proc. of the New Zealand Computer Science Research Student Conference, pp. 49 - 56.

Cui X, Potok TE (2005). Document clustering analysis based on hybrid PSO + KMeans Algorithm. J. Computer Sci. Spl. Iss. pp. 27-33.

Datta S, Giannella CR, Kargupta H (2009). Approximate distributed k-means clustering over P2P network. IEEE trans. on Knowl. Data Eng. 21(10):1372-1388.

Eshref J, Hans-Peter K, Martin P (2003).Towards effective and efficient distributed clustering. Workshop on Clustering Large Data Sets. pp. 1-10.

Hu Z, Zhu W, Li Y E, Du X, Yan F (2013). A Fuzzy approach to clustering of text documents based on MapReduce. Fifth Int'l. Conf. Computat'l. Inf. Sci. (ICCIS). pp. 666-669.

Ibrahim A, Simone A, Ludwig (2012). Parallel particle swarm optimization clustering algorithm based on MapReduce methodology. Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC). pp. 105-111.

Jianxiong Y, Watada J (2011). Decomposition of term-document matrix for cluster analysis. IEEE Int'l. Conf. Fuzzy Systems. pp. 976-983.

Kehua Y, Guoxiong H, Guohui H (2012). Research and application of MapReduce-based MST text clustering algorithm. pp. 753-757.

Khaled MH, Kitchener ON, Kamel MS (2009). Hierarchically distributed peer-to-peer document clustering and cluster summarization. IEEE transac. on Knowl. Data Eng. 21(5):681-698.

Lei Q, Bin W, Qing K, Yuxiao D (2011). SAKU: A distributed system for data analysis in large-scale dataset based on cloud computing. Eighth Int'l. Conf. on Fuzzy Systems and Knowl. Discovery (FSKD). pp. 1257-1261.

Liu P, Ge S (2012). A new distributed name disambiguation system based on MapReduce. IEEE 14th Int'l. Conf. on Commu. Technol. (ICCT). pp. 550-554.

Odysseas Papapetrou, Wolf Siberski, Norbert Fuhr (2012). Decentralized Probabilistic Text Clustering. IEEE Trans. Knowl. Data Engg. 24(10):1848 – 1861.

Patil YK, Nandedkar VS (2014). HADOOP: A New Approach for Document Clustering. Int. J. Adv. Res. in IT and Eng. 3(7):1-8.

Ping H, Jingsheng L, Wenjun Y (2011). Large-Scale Data Sets Clustering Based on MapReduce and Hadoop. J. Computat. Inf. Sys. pp. 5956-5963.

Porter MF (1980). An algorithm for suffix stripping. Program electronic library inf. sys. 14(3):130-137.

Reuters - 21578 text categorization test collection distribution 1.0 retrieved from https://archive.ics.uci.edu/ml/datasets/Reuters21578+Text+Categorization+Collection on September 2014.

Salton G, Wong A, Yang CS (1975). A vector space model for automatic Indexing. Commu. ACM, pp. 613 – 620.

Surendra B, Xian-He S (2011). Special issue on Data Intensive Computing. J. Parallel Distributed Comput. 71(2):143 -144.

Thangamani M, Thangaraj P (2012). Effective fuzzy semantic clustering scheme for decentralised network through multi-domain ontology model. Int. J. Metadata, Semantics Ontol. 7(2):131-139.

Wan J, Yu W, Xu X (2009). Design and implementation of distributed document clustering based on MapReduce. Proc. Symp. Int. Comp. Sci. Comput. Technol. pp. 278-280.

Yang L, Maozhen L, Hammoud S, Khalid AN, Ponraj M (2010). A MapReduce based distributed LSI. Int'l Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 2978-2982.