Journal of Computer Science 7 (5): 762-769, 2011
ISSN 1549-3636

Improving the Performance of Multivariate Bernoulli Model based Documents Clustering Algorithms using Transformation Techniques

[1]Perumal Pitchandi and [2]Nedunchezhian Raju
[1]Department of CSE, Sri Ramakrishna Engineering College, Coimbatore
[2]Department of CSE, Kalaignar Karunanidhi Institute of Technology, Coimbatore

Abstract: Problem statement: Document clustering is the most important areas of data mining since they are very much and currently the subject of significant global research since such areas strengthen the enterprises of web intelligence, web mining, web search engine design and so forth. Generative models based on the multivariate Bernoulli and multinomial distributions have been widely used for text classification. **Approach:** This study explores the suitability of multivariate Bernoulli model based probabilistic algorithm for text clustering application. In a multivariate Bernoulli model, a document is represented as a binary vector over the space of words with 0 and 1, indicating that whether word occurs or not in the document. The number of occurrences is not considered. So the word frequency information is lost due to this nature of implementation. In this work, we propose a FFT based transformation technique for improving clustering performance of multivariate Bernoulli model based probabilistic algorithm. We are using the transformation technique to transform the actual term frequency count data in to a time domain signal. So, the weight of frequency of each word will be distributed throughout each row of records. Now if we apply multivariate Bernoulli model on values less than zero and greater than zero, the performance will get increased since there is no information loss in this kind of data representation. **Results:** In this work, Bernoulli model-based clustering and an improved version of the same will be implemented and evaluated using suitable metrics and the results are shown. **Conclusion:** The transformation technique in multivariate Bernoulli model improves the performance of document clustering significantly.

Key words: Text clustering, text classification, document clustering, model based clustering, term document matrix, Text to Matrix Generator (TMG), Bernoulli model, Fast Fourier Transformation (FFT), transformation technique, clustering algorithms

INTRODUCTION

Clustering: Clustering is the process of grouping a set of physical or abstract objects into classes of similar objects. A cluster is a collection of data objects that are similar to one another with the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group in many applications (Han *et al*., 2011). Clustering is a form of learning by observation rather than learning by examples. Cluster analysis is an important human activity in which we indulge since childhood when we learn to distinguish between animals and plants by continuously improving subconscious clustering schemes. It is widely used in numerous applications, including pattern recognition, data analysis, image processing and market research.

Clustering is a very important application area but widely interdisciplinary in nature, that makes it very difficult to define its scope. It is used in several research communities to describe methods for grouping of unlabeled data, now, these communities have different terminologies and assumptions for the components of the clustering process and the contexts in which clustering is used (Velmurugan *et al*., 2010; Jain *et al*., 1999).Cluster analysis has been studied extensively for years, focusing mainly on distance-based cluster analysis. Many clustering tools were made based on k-means, k-medoids and some of the methods were incorporated in many statistical analysis software packages (Han *et al*., 2011).

Clustering steps:

Preprocessing and feature selection: Most clustering models assume that all data items are represented by n-dimensional feature vectors. This first step therefore

Corresponding Author: P. Perumal, Department of CSE, Sri Ramakrishna Engineering College, Coimbatore
Tel: +919443821151/+918870891194/+914222645786

involves choosing appropriate features and doing appropriate preprocessing and feature extraction on data items to measure the values of the chosen feature set. It will often be desirable to choose a subset of all the features available, to reduce the dimensionality of the problem space. This step often requires a good deal of domain knowledge and data analysis (Suguna *et al*., 2011, Rao, 2003).

Similarity measure: This is a function, which takes two sets of data items as input and returns as output a similarity measure between them. Item-item versions include the weighted lpq norm (and its fuzzy version), the inner product, Hamming distance, Mahalanobis distance and edit distance. Item-set versions use any item-item version as subroutines and include max/min/average distance; another approach looks at the distance from the item to the distance to the cluster representative of the set, where point representatives are chosen as the mean vector/mean center/median center of the set and hyperplane or hyperspherical representatives of the set can also be used. Set-set versions include max/min average distance, as well as item-item versions applied to the two set representatives (Geetha and Kannan, 2007, Adderly, 2002; Jindal, 2006).

Clustering algorithm: Clustering algorithms are general schemes which use particular similarity measures as subroutines. The particular choice of clustering algorithms depends on the desired properties of the final clustering, e.g. what are the relative importance of compactness, parsimony and inclusiveness? Other considerations include the usual time and space complexity (Suguna *et al*., 2011, Rao, 2003).

Result validation: Do the results make sense? If not, we may want to iterate back to some prior stage. It may also be useful to do a test of clustering tendency, to try to guess if clusters are present at all; note that any clustering algorithm will produce some clusters regardless of whether or not natural clusters exist.

Result interpretation and application: Typical applications of clustering include data compression (via representing data samples by their cluster representative), hypothesis generation (looking for patterns in the clustering of data), hypothesis testing (e.g. verifying feature correlation or other data properties through a high degree of cluster formation) and prediction (once clusters have been formed from data and characterized, new data items can be classified by the characteristics of the cluster to which they would belong) (Suguna *et al*., 2011, Rao, 2003).

Document clustering: The document clustering is the core topic in the information retrieval field. It uses unsupervised algorithms to cluster a large amount web page into several groups. Let's take an example to illustrate why document clustering is necessary. Everyone has experienced Google search for information from Internet. In response to a query of a web client, Google will send back tons of web pages. Although they are listed by the order of its importance, users still sometimes have to browse hundreds of web page to find what they want. If we can group the pages into groups, users can skip the group they are not interested in. They will not have to browse too many pages before reaching their targets. This will help users to do their queries efficiently. However the problem is How to group pages? The answer is using document clustering.

Key requirements for document clustering are:

- How to present a document in the mathematical model
- Different kinds of document cluster algorithms
- Some refinements to the clustering algorithm
- How to choose an appropriate topic to present the clusters
- How to evaluate the algorithms and resulting clusters
- Evaluate and compare different algorithms
- A real world document clustering application

Document clustering has become an increasingly important technique for unsupervised document organization, automatic topic extraction and fast information retrieval or filtering [Performance].

In many emerging data mining applications, one needs to cluster complex data such as very high-dimensional sparse text documents and continuous or discrete time sequences. Probabilistic model-based clustering techniques have shown promising results in many such applications. For real-valued low-dimensional vector data, Gaussian models have been frequently used. For very high-dimensional vector and non-vector data, model-based clustering is a natural choice when it is difficult to extract good features or identify an appropriate measure of similarity between pairs of data objects (Suguna *et al*., 2011, Zhong, 2003).

The vector space model and document clustering: Generally in document clustering algorithms,

documents are represented using the vector-space model. In this model, each document, d, is considered to be a vector, d, in the term-space (set of document "words"). In its simplest form, each document is represented by the Term Frequency (TF) vector:

$d_{tf} = (tf_1, tf_2, \ldots, tf_n)$

where, tf_i is the frequency of the i^{th} term in the document (Hyma *et al*., 2010).

Problem definition: The database of document is represented as term document matrix in which each column will represent a word in one or more documents and each row will represent a text document. Each number in a row will represent the number of counts of different words in that document. So practically, in a typical Term Matrix, there will be lot of zeros that will represent the absence of a word in a document. Further, there will be huge values representing frequent assurance of some particular words in most of the documents. So the magnitudes of these individual values will be scattered between zeros to a very large number.

So even though this is a multidimensional numerical data set, any typical clustering algorithm will not create meaningful clusters out of them because of the waste differences in word counts. The distance metrics generally used for finding distance between record sets in a clustering algorithm will fail to find exact distance between these document vectors.

In an earlier multivariate Bernoulli model (Sumathi *et al*., 2010, Zhong and Ghosh, 2003), a document is represented as a binary vector over the space of words. The dimension of a document vector is denoted by either 0 or 1, indicating whether word w_l occurs or not in the document. Even though their model proposed better results than classical distance based approaches like normal k-means algorithm, still there was some performance lack due to the binary form of representation of data. In this representation, the number of occurrences is not considered, so the word frequency information is lost.

In some of the other the earlier implementations of the algorithms namely, Multinomial model-based clustering, von Mises-Fisher model-based clustering, they did not used the actual frequency counts in the calculations.

So in our implementation we will do little changes in that algorithm and will use the transformed virtual time domain representation of document data instead of actual data.

The solution strategy: To make the distance metric to work better on document data, we are going to convert the two dimensional term frequency data in to a time domain signal. For that, we propose a method to handle this situation using Fourier Transformation Technique (FTT). By assuming the two dimensional term frequency matrix data as a transformed data, inverse Fourier transformation is applied to get a hypothetical original data.

That is, the actual data is considered as a frequency domain representation of the documents and the time domain signals are derived from that using inverse Fourier transformation.

This study explores the suitability of multivariate Bernoulli model based probabilistic algorithm for text clustering application. In a multivariate Bernoulli model, a document is represented as a binary vector over the space of words with 0 and 1, indicating whether word occurs or not in the document. The number of occurrences is not considered. So the word frequency information is lost due to this nature of implementation.

We are using this transformation technique to transform the actual term frequency count data into a time domain signal. So, the weight of frequency/word count of each word will be distributed throughout all columns of each row of records. Now if we apply multivariate Bernoulli model probability calculations on values less than zero and greater than zero, the performance will get increased since there is no information loss in this kind of data representation.

Evaluating the quality of the cluster results: Validating clustering algorithms and comparing performance of different algorithms is complex because it is difficult to find an objective measure of quality of clusters. In order to compare results against external criteria, a measure of agreement is needed. Since we assume that each record is assigned to only one class in the external criterion and to only one cluster, measures of agreement between two partitions can be used (Dalton *et al*., 2009).

Purity is a simple and transparent evaluation measure. Normalized mutual information can be information-theoretically interpreted. The Rand index penalizes both false positive and false negative decisions during clustering (Dalton *et al*., 2009).

Purity measure: To compute purity, each cluster is assigned to the class which is most frequent in the cluster and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by N Formally:

$$\text{Purity}(\Omega, C) = \frac{1}{N}\sum_{k}\max_{j} | \omega_k \cap c_j |$$

Where:

$\Omega = \{\omega_1, \omega_2,\ldots,\omega_k\}$ = The set of clusters
$C = \{c_1, c_2,\ldots,c_J\}$ = The set of classes

We interpret ω_k as the set of documents in ω_k and c_j as the set of documents in c_j the above Equation.

High purity is easy to achieve when the number of clusters is large-in particular, purity is 1 if each document gets its own cluster. Thus, we cannot use purity to trade off the quality of the clustering against the number of clusters.

Mutual information measure: The mutual information I(X; Y) between a random variable X, governing the cluster labels and a random variable Y, governing the class labels, is a superior measure than purity or entropy (Sumathi *et al*., 2010, Zhong and Ghosh, 2003). Moreover, by normalizing this measure to lie in the range [0, 1]; it becomes quite impartial to k. There are several choices for normalization based on the entropies H(X) and H(Y). We shall follow the definition of Normalized Mutual Information (NMI) using geometrical mean, $\text{NMI} = \frac{I(X;Y)}{\sqrt{H(X).H(Y)}}$ as given in (Sumathi *et al*., 2010, Zhong and Ghosh, 2003). In practice, we use a sample estimate:

$$\text{NMI} = \frac{\sum_{h,l} n_{h,l} \log\left(\frac{n.n_{h,l}}{n_h n_l}\right)}{\sqrt{\left(\sum_h n_h \log\frac{n_h}{n}\right)\left(\sum_l n_l \log\frac{n_l}{n}\right)}}$$

where, n_h is the number of data samples in class h, n_l the number of samples in cluster l and n_h; l.

The number of samples in class h as well as in cluster l. The NMI value is 1 when clustering results perfectly match the external category labels and close to 0 for a random partitioning.

Rand index: The Rand index or Rand measure is a commonly used technique for measure of such similarity between two data clusters.

Given a set of n objects S = {O1... On} and two data clusters of S which we want to compare: X = {x1... xR} and Y = {y1... yS} where the different partitions of X and Y are disjoint and their union is equal to S; we can compute the following values (Dalton *et al*., 2009):

- a is the number of elements in S that are in the same partition in X and in the same partition in Y
- b is the number of elements in S that are not in the same partition in X and not in the same partition in Y
- c is the number of elements in S that are in the same partition in X and not in the same partition in Y
- d is the number of elements in S that are not in the same partition in X but are in the same partition in Y

Intuitively, one can think of a + b as the number of agreements between X and Y and c + d the number of disagreements between X and Y. The rand index, R, then becomes:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$

The rand index has a value between 0 and 1 with 0 indicating that the two data clusters do not agree on any pair of points and 1 indicating that the data clusters are exactly the same.

Probabilistic model based document clustering: The traditional vector space representation is used for text a document, i.e., each document is represented as a high dimensional vector of "word" counts in the document. The dimensionality equals the number of words in the vocabulary used.

In k-means, we attempt to find k centroids that are good representatives. We can view the set of k centroids as a model that generates the data. Generating a document in this model consists of first picking a centroid at random and then adding some noise. Model-based clustering assumes that the data were generated by a model and tries to recover the original model from the data. The model that we recover from the data then defines clusters and an assignment of documents to clusters.

The model can be adapted to what we know about the underlying distribution of the data, be it Bernoulli, Gaussian with non-spherical variance (another model that is important in document clustering) or a member of a different family.

Model-based k-means: The model-based k-means (mk-means) algorithm is a generalization of the standard k-means algorithm, with the cluster centroid vectors being replaced by probabilistic model. Let X = $\{x_1,\ldots,x_N\}$ be the set of data object and ^ = $\{\lambda_l,\ldots, \lambda_k\}$ the set cluster models. A commonly used criterion for

estimating the model parameters is maximum likelihood. The mk-means algorithm locally maximizes the log-likelihood objective function:

$$\log P(X \mid \wedge) = \sum_{x \varepsilon X} \log p(x \mid \lambda_{y(x)})$$

where, $y(x) = \arg \max_y \log p (x|\lambda_y)$ is the cluster identity of object x.

So a generic model based algorithm will have following steps:

- Initialize the cluster identity vector
- Model re-estimation step
- Sample re-assignment step
- Convergence Check and repeat from 2 or stop

Multivariate Bernoulli model: In a multivariate Bernoulli model (Geetha and Kannan, 2007; Zhong and Ghosh, 2003), a document is represented as a binary vector over the space of words. The l-th dimension of a document vector x is denoted by x (l) and is either 0 or 1, indicating whether word w_l occurs or not in the document. The number of occurrences is not considered, i.e., the word frequency information is lost.

With naïve Bayes assumption, the probability of a document x in cluster y is:

$$P(x \mid \lambda_y) = \prod_l Py(w_l)_{x(l)}(1 - Py(w_l))_{1-x(l)}$$

where, $\lambda y = \{P_y(w_l)\}$, $P_y(w_l)$ is the probability of word w_l being present in cluster y and $(1\text{-}P_y(w_l))$ the probability of word w_l not being present in cluster y. To avoid zero probabilities when estimating $P_y(w_l)$, one can employ the solution as (Geetha and Kannan, 2007; Zhong and Ghosh, 2003):

$$P_y(w_l) = \frac{1 + \sum_x P(y \mid x, \wedge) x(l)}{2 + \sum_x P(y \mid x, \wedge)}$$

where, P(y| x, ^) is the posterior probability of cluster y.

In the above model, the data is reduced to binary form in probability calculations. If a particular word is not found in a document, then it is represented as 0. On the other hand, if it is present then it is represented as 1 irrespective of the number of occurrences of the word in that document. This will lead to inaccuracy in calculations. To avoid this, first we are transforming the data using Fourier transformation.

MATERIALS AND METHODS

The Fourier Transform is based on the discovery that it is possible to take any periodic function of time x (t) and resolve it into an equivalent infinite summation of sine waves and cosine waves with frequencies that start at 0 and increase in integer multiples of a base frequency $f_0 = 1/T$, where T is the period of x (t) (Duhamel and Vetterli, 1990). Here is what the expansion looks like:

$$x(t) = a_0 + \sum_{k=1}^{\infty} (a_k \cos(2\pi k f_0 t) + b_k \sin(2\pi k f_0 t))$$

An expression of the form of the right hand side of this equation is called a Fourier series. Fast Fourier Transformation (FFT) is an algorithm to compute the Discrete Fourier Transform (DFT) even with computers with limited computing power.

Take one document in the dataset. Let the document vector V = {f1, f2, f2 …. Fn} represents that one document in the database. Where f1, f2, f2 …. Fn are the n word counts/frequencies representing that document. If we consider this as a frequency domain representation of that document (in signal processing we will get this kind of frequency domain data if we transform the time domain signal) then we can apply inverse Fourier transform to estimate the imaginary time domain representation of that document.

So the function FFT (V) will give a time domain signal where all the weights of any frequency/word count. Generally FFT calculation will result both imaginary part as well as real part. We can only consider real part of the output for our clustering calculations.

For example, if the first histogram in the following Fig. 1 graphically represents the word counts (magnitude in y axis) of 130 words (x-axis) in a document, then the inverse Fourier transformation of that frequency domain signal will give the second signal which will be the indirect time domain representation of the document (here we can represent time as imaginary locations of words in the document) starting from 1 to 200-200 word locations/word in that document).

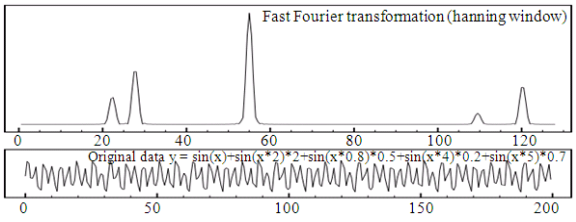

Fig. 1: The frequency domain and time domain representation of the document

If we observe the time domain signal, the distribution of amplitude is almost even-so any distance calculation function will give better result.

Further in Bernoulli model, during calculating the log-likelihood, the < 0 values and the values between 0 and 1 will be handled separately. But, if we use FFT transformed data, then there will be chance of getting high negative values as well as high positive values. So in the proposed method, during calculating the log-likelihood, the < 0 values and the values > 0 or >1 may be handled separately.

RESULTS

To evaluate the algorithms, a suitable and standard data set is needed. We decided to use some of the same datasets which were originally used in a previous reference work (Perumal and Nedunchezhian, 2011; Zhong and Ghosh, 2003). The datasets were originally from TREC collections (http://trec.nist.gov). Datasets tr11, tr23, tr41 and tr45 were originally derived from TREC-5, TREC-6 and TREC-7 collections. (NIST Text REtrieval Conferences-TREC). We used TMG format of these datasets which is available in several internet resources.

Accuracy of clustering with different datasets: Upon using FFT the values obtained are tabulated with respect to Normal Bernoulli model and improved Bernoulli model in terms of Rand Index as a metric, from Table 1 it can be observed that upon computing the average for both the models, the proposed improved Bernoulli model gives a better result in comparison to the normal Bernoulli model.

DISCUSSION

The tabulated values of Table 1 are represented as bar chart in Fig. 2. The x-axis consist of the datasets: Tr11, Tr12, Tr23, Tr31 and Tr41 and the y-axis as a Rand Index value. The normal Bernoulli model and improved Bernoulli model for each data set can be inferred in this chart and it clearly shows that for all the dataset values taken, improved Bernoulli model provides better results.

Upon using FFT the values obtained are tabulated with respect to Normal Bernoulli model and improved Bernoulli model in terms of Mutual information measure as a metric, from Table 2 it can be observed that upon computing the average for both the models, the proposed improved Bernoulli model gives a better result in comparison to the normal Bernoulli model.

The following Fig. 3 shows results based on Table 2. It clearly shows that improved Bernoulli model gives better results than normal Bernoulli model.

Upon using FFT the values obtained are tabulated with respect to Normal Bernoulli model and improved Bernoulli model in terms of Mutual information measure as a metric, from Table 3 it can be observed that upon computing the average for both the models, the proposed improved Bernoulli model gives a better result in comparison to the normal Bernoulli model.

The above Fig. 4 shows results based on Table 3. It clearly shows that improved Bernoulli model gives better results than normal Bernoulli model.

Table 1: Accuracy in terms of rand index with different data sets

	Clustering accuracy in terms of rand index (average of three runs)	
Data set used	Normal Bernoulli model	Improved Bernoulli model
Tr11 414×6424	0.44350	0.86040
Tr12 313×5799	0.42180	0.83230
Tr23 204×5831	0.36120	0.72310
Tr31 927×10127	0.54600	0.76850
Tr41 690×8261	0.58310	0.87690
Avg	0.47112	0.81224

Table 2: Accuracy in terms of mutual information measure with different data sets

	Clustering accuracy in terms of mutual information (average of three runs)	
Data set used (Name Row x Cols)	Normal Bernoulli model	Improved Bernoulli model
Tr11 414×6424	0.20770	0.52590
Tr12 313×5799	0.23120	0.49170
Tr23 204×5831	0.24730	0.38310
Tr31 927×10127	0.20230	0.42660
Tr41690×8261	0.19780	0.63000
Avg	0.21726	0.49146

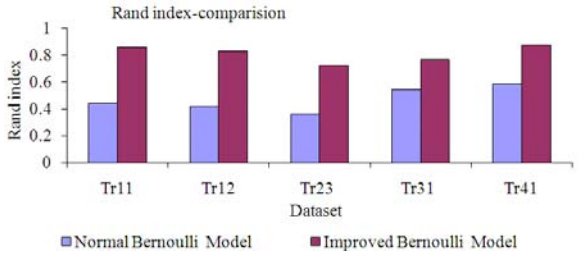

Fig. 2: Accuracy chart-rand index

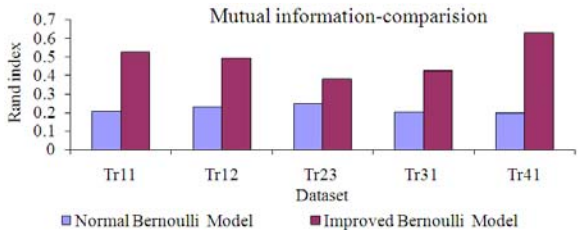

Fig. 3: Accuracy chart-mutual information measure

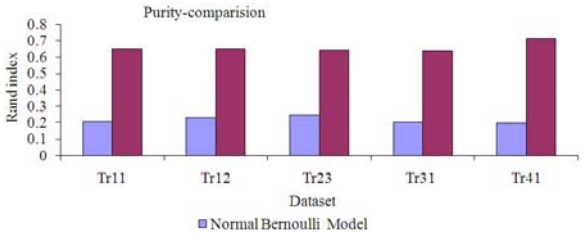

Fig. 4: Accuracy chart-purity measure

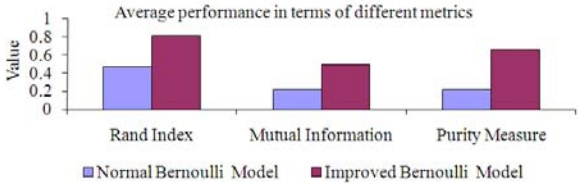

Fig. 5: Comparison of accuracy with different metrics

Table 3: Accuracy in terms of purity with different data sets

	Clustering accuracy in terms of mutual information (average of three runs)	
Data set used (Name Row x Cols)	Normal Bernoulli model	Improved Bernoulli model
Tr11 414×6424	0.20770	0.65320
Tr12 313×5799	0.23120	0.65050
Tr23 204×5831	0.24730	0.64460
Tr31 927×10127	0.20230	0.64010
Tr41 690×8261	0.19780	0.71450
Avg	0.21726	0.66058

Table 4: The average performance with respect to different metrics

	Clustering accuracy in terms of mutual information (average of three runs)	
Metric	Normal Bernoulli model	Improved Bernoulli model
Rand index	0.47112	0.81224
Mutual information measure	0.21726	0.49146
Purity measure	0.21726	0.66058

The Table 4 shows that all three metrics of normal Bernoulli model and improved Bernoulli model. It clearly shows that in all three metrics, the improved Bernoulli model gives better results than normal Bernoulli model.

The above Fig. 5 shows that the comparison of accuracy with different metrics in terms of bar chart. From the bar chart, it can be inferred that improved Bernoulli model gives better result in terms of all three metrics.

CONCLUSION

In the proposed work the Multivariate Bernoulli Model has been explored and it has been observed that, the performance of this work is better. In comparing the performance of the two versions of the algorithms in terms of Mutual Information Measure, Purity Measure and Rand Index, the performance of the modified version of algorithm was very much better than the normal Multivariate Bernoulli Model based clustering. Further all the three metrics clearly signified the difference in performance.

It is also found that if binary representation is used to represent document as vectors in a Term-Document Matrix then there is a large difference in magnitude of individual attributes of data. To overcome these drawbacks, FFT based data transformation technique with changes in Bernoulli model has been proposed to achieve better accuracy in clustering. Future works may be extended to find efficient computation methods to minimize the time complexity involved in large datasets.

ACKNOWLEDGEMENT

We thank to Director, Principal and the management of Sri Ramakrishna Engineering College for providing lab facility to implement this work.

REFERENCES

Adderly, M.D., 2002. Data mining meets e-commerce: using data mining to improve customer relationship management. Thesis, Master of Science, University of Florida, pp: 1-69. http://etd.fcla.edu/UF/UFE0000500/adderly_d.pdf

Dalton, L., V. Ballarin and M. Brun, 2009. Clustering algorithms: On learning, validation, performance and applications to genomics. Curr. Genomics, 10: 430-445. DOI: 10.2174/138920209789177601

Duhamel. P and Vetterli. M, 1990. Fast Fourier transforms: A tutorial review and a state of the art. Signal Process., 19: 259-299. DOI: 10.1016/0165-1684(90)90158-U

Geetha, A. and A. Kannan, 2007. Enhancement of search results using dynamic document seed reranking algorithm. J. Comput. Sci., 3: 436-440. http://www.doaj.org/doaj?func=abstract&id=419087

Han, J., K. Micheline and J. Pei, 2011. Data Mining: Concepts and Techniques. 2nd Edn., Morgan Kaufmann Publishers, San Francisco, USA., ISBN: 1558609016, pp: 770.

Hyma, J. *et al*., 2010. A new hybridized approach of PSO and GA for document clustering. Int. J. Eng. Sci. Technol., 2: 1221-1226. http://www.ijest.info/docs/IJEST10-02-05-99.pdf

Jain, AK., M.N. Murty and P.J. Flynn, 1999. Data clustering: A review. ACM Comput. Surveys, 31: 264-323. DOI: 10.1145/331499.331504

Jindal, R., 2006. An empirical clustering techniques. Thesis, Master of Engineering, University of Delhi, pp: 1-84.

Perumal, P. and R. Nedunchezhian, 2011. Performance evaluation of three model-based documents clustering algorithms. EJSR, 52.

Rao, R., 2003. Data mining and clustering techniques. Proceedings of the DRTC Annual Workshop on Semantic Web, Dec. 8-10, Bangalore, Paper K., pp: 1-12.

Suguna, N. and K.G. Thanushkodi, 2011. Predicting missing attribute values using k-means clustering. J. Comput. Sci., 7: 216-224. DOI: 10.3844/jcssp.2011.216.224

Sumathi, C.P., R. P.Valli and T. Santhanam, 2010. An application of session based clustering to analyze web pages of user interest from web log files. J. Comput. Sci., 6: 785-793. DOI: 10.3844/jcssp.2010.785.793

Velmurugan, T. and T. Santhanam, 2010. Computational complexity between k-means and k-medoids clustering algorithms for normal and uniform distributions of data points. J. Comput. Sci., 6: 363-368. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.165.9474&rep=rep1&type=pdf

Zhong, S. and Ghosh, J. 2003. A comparative study of generative models for document clustering. The University of Texas at Austin. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.8.4583&rep=rep1&type=pdf

Zhong, S., 2003. Probabilistic model-based clustering of complex data. PhD Thesis, The University of Texas at Austin. http://portal.acm.org/citation.cfm?id=997792