CONTENT BASED SPARK-ITFS FEATURE SELECTION FOR EXTRACTING USEFUL INFORMATION IN BIG DATA

Karthick N[1], X. Agnes Kalarani[2]

ABSTRACT

Big data handling is the most important challenges faced by many of the researchers in the world due to its varying structure and high volume of contents. The most useful and relevant information plays the most important role in the real world application environment scenario, which decides the successful completion of the task execution. Finding the useful information from the big data which consist of more irrelevant data would be the most complex process which needs to be done with more care for designing the most flexible framework that can handle the large volume of data in an efficient manner. Filtering is one of the most popular approaches, which is frequently followed by most of the researchers for eliminating the irrelevant columns and retrieving only useful information. There are various filtering mechanisms such as low variance filter, highly correlated filter, PCA filter are introduced in the existing scenarios for filtering the irrelevant information. However these mechanisms cannot perform well in the case of large volume of data where the data growth is more in run time. In this proposed research methodology, this problem is resolved by introducing the Content based Spark-ITFS feature selection approach on big data. The proposed research methodology can better filter out the irrelevant data columns that are present in the database in the distributed and accurate manner. The experimental evaluation was conducted in the Hadoop Simulation environment by using KDD cup 99 dataset. The findings of this work demonstrate that the proposed research methodology Content based Spark-ITFS feature selection approach can perform better than the existing research methodologies in terms of improved accuracy, precision, recall and F-Measure values.

Keywords : Big data handling, Filtering, Relevant feature selection, Useful information, Partitioning

1. INTRODUCTION

Big data is a large volume of data set which is growing dynamically in run time due to increased real world applications. Big data plays an important role in the real world organizations by providing them the useful information for completing their tasks. Big data are denoted as 3 v because of their properties such as volume (Increased volume of data dynamically), Velocity (Speed of growing data size) and Variety (Different types of data extracted from

[1]Research Scholar, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore - 641021, Tamil Nadu, India. Corresponding author E-mail : karthickphd333@gmail.com

[2]Professor, Department of Computer Application, Karpagam Academy of Higher Education, Coimbatore – 641021, Tamil Nadu, India. E-mail : agneskala72@gmail.com

different types of source databases). These properties of big data create more complex environment for the researchers from getting the information that are relevant to perform their tasks. Finding and retrieving the relevant data from large volume of data would be a more difficult process.

Data dimensionality reduction is one of the best known techniques which are followed frequently by many of the researchers to reduce the dimensionality of the data sets that is collected from various data sources. There are various techniques available in the real world environment for reducing the dimensionality of the data sets in an efficient manner. All of the technologies attempt to filter out the irrelevant columns or reduce the more relevant columns into one. These data dimensionality reduction approaches can perform filtering process efficiently than other tasks, thus the big data handling management would be done more efficiently. However the large volume of data set and its growing nature would lead to the failure in big data handling management due to its varying nature and its structure. This scalability issue needs to be resolved efficiently to make ease of big data handling management tasks.

The main objective of this research work is the utilization of big data efficiently by selecting relevant information from the big data by omitting the irrelevant data. The overall goal of this work is to improve the big data oriented application performance by filtering out the more relevant data from the big data with reduced network traffic cost. The overall research of this work attempts to design a big data handling management framework that can efficiently handle the large volume of data in the considerable manner. The overall research of this work is implemented in the Hadoop simulation environment and compared with the various previous works in terms of performance measures called the accuracy, precision, recall and f-measure.

The overall organization of the research work is given as follows: In the section 2, previous research works that have been conducted to filter out the irrelevant information are given. In section 3, proposed research of this methodology is discussed in detail. In section 4, performance evaluation of the research methodology is discussed in a detailed manner. In section 5, the overall conclusion of this work is given.

2. Related Works

In this section, various research works that are conducted in terms of different performance measures are discussed in a detailed manner.

Zhe Wang et al [2010] propose a novel approach to fuse a tag-based neighborhood method into the traditional rating-based CF. Tag-based neighborhood method is employed to find the similar users and items. This neighborhood information helps the sequent CF procedure to produce the higher quality recommendations. The approach is compared with two collaborative filtering algorithms Non-negative Matrix Factorization (NMF) method, PMF method and the improved regularized SVD method. The main disadvantage is that, if there are a lot of new entries from the users to items, the proposed work solution fails to deal with this situation.

Ning Zhou et al [2011] presented a novel hybrid probabilistic model (HPM) which integrates low-level image features and high-level user provided tags to automatically tag images. For images without any tags, HPM predicts new tags based on the low level image features. For images with user provided tags, HPM jointly exploits both the image features and the tags in a unified probabilistic framework to recommend additional tags to label the images. This paper proposes a collaborative filtering method based on nonnegative matrix factorization (NMF) for tackling data sparsity issue

Mohamed Amine Chatti et al [2013] proposed 16 different tag-based collaborative filtering recommendation algorithms, that are memory based as well as model based. This paper also says different tag-based collaborative filtering recommendation techniques on their applicability and effectiveness in PLE settings. The main drawback in the system is that this process required additional experimental tuning of the parameters of all used data mining techniques. The results of the evaluation in this experiment revealed that the Item-Based k-Means clustering was the best performing algorithm in the offline evaluation whereas the user based Apriori algorithm was ranked first in the user evaluation. Generalizing these two algorithms is not encouraged because this paper uses only a small sample size.

Feng Xie et al [2013] proposed a method to eliminate data scarcity problem. Inaccurate similarities derived from the sparse user-item associations would generate the inaccurate neighborhood for each user or item. Consequently, its poor recommendation drives us to propose a Threshold based Similarity Transitivity (TST) method in this paper. TST firstly filters out those inaccurate similarities by setting an intersection threshold and then replaces them with the transitivity similarity. Obviously, when the threshold is set to 6, TST does not achieve the best performance in coverage and popularity. Analogously, TST gets lower coverage and higher popularity with threshold 3 than some of the other thresholds.

Rong Hu et al [2013] proposed a clustering-based collaborative filtering approach, which aims at recruiting similar services in the same clusters to recommend services collaboratively. Technically, this approach is enacted around two stages. In the first stage, the available services are divided into small-scale clusters for further processing. At the second stage, a collaborative filtering algorithm is imposed on one of the clusters. The major disadvantage of the paper is that it does not include service similarity.

Shanshan Yao et al [2015] proposed a multi-stage filtering strategy which can achieve both speedup and high accuracy, with the beginning stages focusing on speedup and the end stage emphasizing accuracy. With this idea, an efficient cascaded filtering retrieval method is proposed that consists of filtering with Fibonacci hashing. The middle fingerprint can quickly filter out most irrelevant audios without sacrificing accuracy when applied to filtering other than refining. The main drawback is that the classification algorithm is not so effective in improving the discrimination ability of the middle finger.

In Juan J. Pomárico-Franquiz, et al [2015] the extended Kalman filter (EKF) algorithm is modified and a new extended unbiased finite impulse response (EFIR) filtering algorithm is developed. In RFID networking systems, target state can be observed over a big number of tags. It is also shown that target state observation over the RFID tag excess channels allow mitigating effect of the imprecisely defined noise statistics on the EKF performance and preventing divergence in EKF. The main drawback is that when there are deviations from the actual noise co variances in nonlinear state-space models may cause the same divergence in EKF.

Holly Alexander et al [2015] proposed a novel cache sharing system employing data structures called Dynamic Interest-Tagged Filtered Bloom Filters (DITFBFs). DITFBFs are capable of determining the cache content of a proxy in a compact form, which is then shared with other proxies in the cooperative Web caching system. The proposed system is different from others, as DITFBFs is the only one that represents the portion of a proxy's cache content that will be of interest to another proxy. This results in a reduction of inter-proxy overhead. Because the proposed system is designed to work within environments that engender well-defined and distinct interests, it has a drawback that its performance will naturally begin to degrade in environments that don't produce such interests. It is possible that, over time, the proposed system will be subject to interest drift that is interests may become fewer representatives as time goes by unless they are consistently updated.

3. Spark-Itfs For Distributed Filtering On Big Data

Filtering is the most challenging task in the big data environment where the large volume of data exists with different form. This filtering process is used to reduce the dimensionality of the data set by eliminating the more irrelevant data columns that are present in the data set. There are various previous research work that are introduced by various authors for performing filtering process on data set. Some of the previous filtering techniques are low variance filter, High correlation filter, and PCA filter. These filters attempt to reduce the dimensionality of the data set by eliminating the data columns that are irrelevant to the task. These filters process well in the small data set which is degraded in its performance in case of presence of big data with dynamic growing nature.

In this work, filtering on big data is performed efficiently by introducing the methodology called the Spark-ITFS. This filtering approach works with the consideration of the relevancy of columns and as well as redundancy of the columns. This filtering methodology can perform well in the presence of the big data with dynamic growing nature and it can establish the contents that are well representing the useful information. This spark-ITFS is a distributed algorithm which can establish the filter out the irrelevant columns in the iterative until convergence achieved. This distributed nature of the Spark-ITFS enables flexible big data handling management system with the supportive of run time growth of the

data. Spark ITFS is applied on content based partitioning approach to develop Content Partitioning based Spark-ITFS

The detailed explanation of various previous research works and the proposed research methodology are discussed in detail in the following sub sections.

Low Variance Filter

Low variance filter performs better filtering process by eliminating the unwanted columns that is irrelevant data columns that are present in the data base. It will calculate variance between the columns present in the big data, and it will be compared with threshold value. The columns with less variance value than the threshold value would be ignored from the database by considering it as irrelevant data column. The variance is calculated as follows:

$$\sigma^2 = \frac{\sum(X-\mu)^2}{N}$$

Where

X → Set of data

μ → mean value

N → Total number of data

Highly Correlated Filter

Highly correlated filter perform filtering process by replacing the highly correlated data columns by others. This is done by calculating the correlation between the data columns present in the data set. After finding the correlation, the highly correlated data column would be eliminated instead of keeping all. This is the better approach than the low variance filter which attempts to keep all unique columns where the low variance filter would eliminate the unique columns. Simple correlation is used to identify the strength of correlation between different data columns. Correlation is calculated as follows

$$r = \frac{n\sum xy - (\sum x)(\sum y)}{\sqrt{n(\sum x^2) - (\sum x)^2}\sqrt{n(\sum y^2) - (\sum y)^2}}$$

Where

r → correlation value

x → data column 1 values

y → data column 2 values

Principal Component Analysis Filter

PCA filter performs better filtering process by converting the original data values into orthogonal data values which would be computed for data correlation. PCA can filter out more relevant data columns and will keep retaining most of the data information. The PCA technique would compute the covariance matrix in which covariance of data column values would be stored along with its Eigen vector values. Then the correlation will be calculated for the Eigen values of the Eigen vectors in terms of their converted values. The PCA model produced at the last output port of the PCA Compute node contains the Eigen values and the eigenvector projections necessary to transform each data row from the original space into the new PC space. The PCA Apply node transforms a data row from the original space into the new PC space, using the eigenvector projections in the PCA model. A point from the original data set is converted into the new

set of PC coordinates by multiplying the original zero-mean data row by the eigenvector matrix from the spectral decomposition data table. By reducing the number of eigenvectors, the dimensionality of the new data set is efficiently reduced. Based on the statistical distribution of the data set that is fed into the PCA Compute node, the PCA Apply node calculates the dimensionality reduction (how many of the input features are kept) with respect to the proposed information (variance) preservation rate.

Content Partitioning based Spark-ITFS

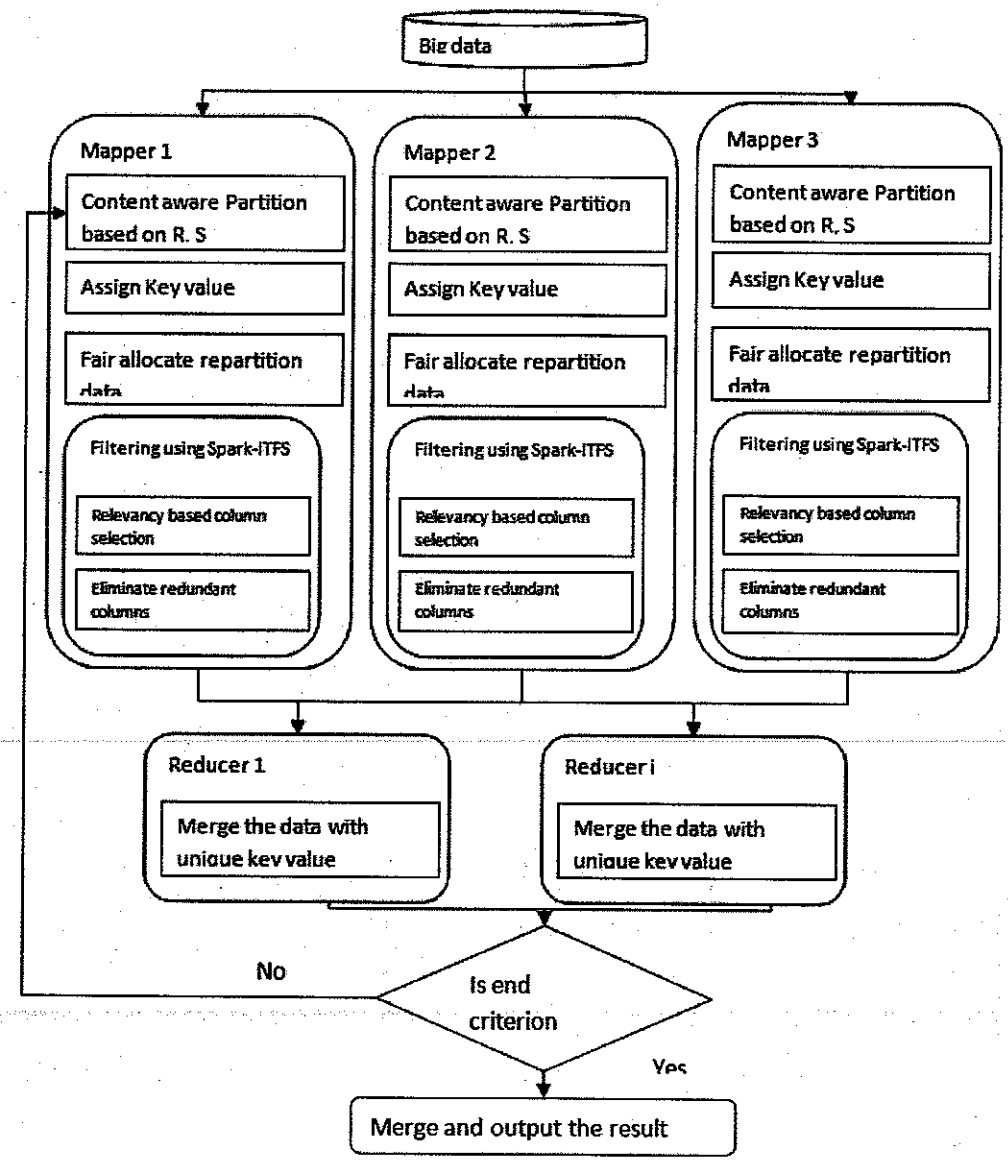

Figure 1 : Content based spart-IFTS feature selection

Content-aware partition method partitions dataset by identifying the appropriate partitioning columns and the size of the partition columns. By using this information partitioning would be done which will generate different partitioning data columns that are irrelevant to each other. Thus the data linkage problem can be avoided considerably. After content aware partitioning, the fair-allocation block placement is utilized to allocate the blocks to the appropriate data nodes. Then the SPARK-ITFS filter based feature selection process is carried out to effectively reduce the dimensionality of data in each of the nodes. SPARK_ITFS functions in the distributed manner, where each feature would be handled in the single iteration. SPARK_ITFS follows two major steps for better selection of features from the big data in the multiple iteration. Those are

→ Relevancy calculation

→ Redundancy finding

Initially, relevancy between the features of a class would be calculated by using the following equation

$$I(A;B) = H(A) - H(A|B) = \sum_{a\in A}\sum_{b\in B} p(a|b)\log\frac{p(a|b)}{p(a)p(b)}$$

The more relevant features would be considered for further processing which will be cached. After that, the redundancy values are calculated between the non-selected features and the last one is selected.. The redundancy is calculated as follows:

$$I(A;B|C) = H(A|C) - H(A;B|C) = \sum_{c\in C} p(c) \sum_{a\in A}\sum_{b\in B} p(ab|c)\log\frac{p(ab|c)}{p(a|c)p(b|c)}$$

This process would be repeated for reducing the dimensionality by eliminating the irrelevant features from the data set. The work flow of this methodology is given as in figure 1.

4. Experimental Results

This section describes the performance evaluation which is conducted between the proposed methodologies along with the various existing methodologies in the Hadoop simulation environment. The KDD cup data set is used to extract the useful information and compare the effectiveness of our algorithm. This is the data set used for The Third International Knowledge Discovery and Data Mining Tools Competition, which was held in conjunction with KDD-99, The Fifth International Conference on Knowledge Discovery and Data Mining. The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between "bad" connections, called intrusions or attacks, and "good" normal connections. This database contains a standard set of data to be audited, which include a wide variety of intrusions simulated in a military network environment.

The comparison is made between the existing methodologies and the proposed methodologies namely Low variance filter, High correlation filter, PCA Filter, Content partitioning based Spark-ITFS in terms of accuracy, precision, recall and F-measure. This is described in detail in the following sections.

Accuracy

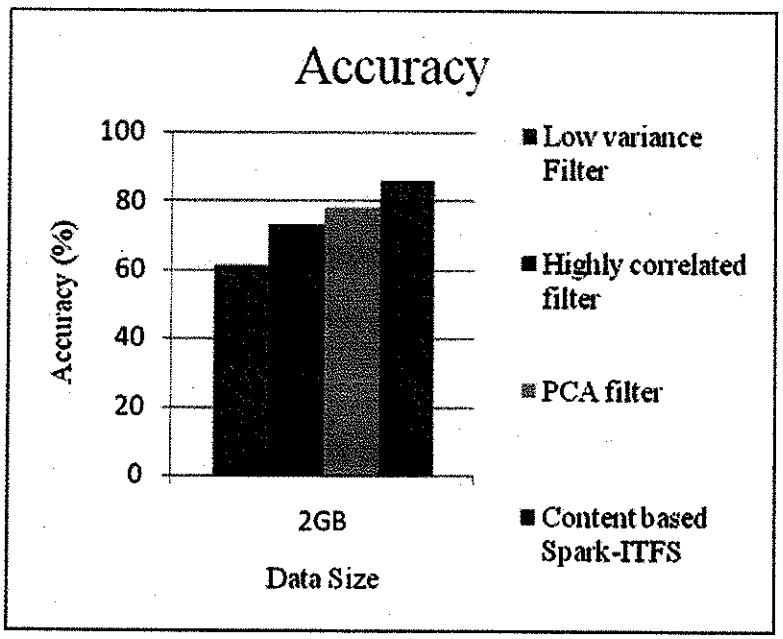

Figure 2 : Accuracy comparison

Accuracy is defined as the proportion of true positives and true negatives among the total number of results obtained. Accuracy is evaluated as,

$$Accuracy = \frac{(True\ positive + True\ negative)}{(True\ positive + True\ negative + False\ positive + False\ negative)}$$

From the Figure 2 it is proved that the Content partitioning based spark-ITFS can filter efficiently in terms of better accuracy. From the above graph, it is shown that the Low variance filter provides 61 % accuracy rate, Highly correlated filter provides 73 % accuracy rate, PCA filter provides 78 % accuracy rate, content based Spark-ITFS provides 86% accuracy rate. From this it is proved that Content partitioning based spark-ITFS provides better result than the other methodologies.

Precision

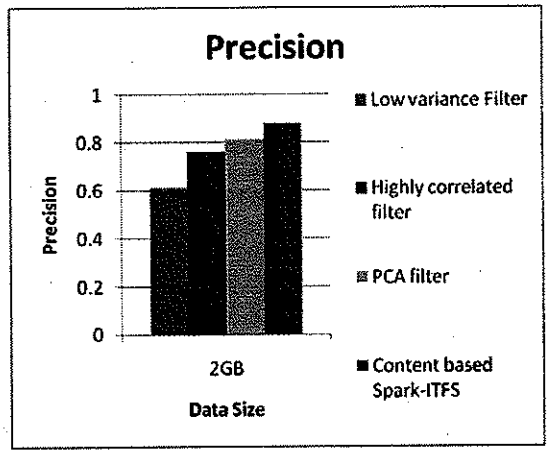

Figure 3 : Precision Comparison

Precision value is evaluated according to the relevant information at true positive prediction, false positive.

$$Precision = \frac{True\ positive}{(True\ positive + False\ positive)}$$

From the figure 3 it is proved that the Content partitioning based spark-ITFS can filter efficiently in terms of better precision. From the above graph, it is shown that the Low variance filter provides 0.61 precision values, Highly correlated filter provides 0.76 precision value, PCA filter provides 0.81 precision value, and content based Spark-ITFS provides 0.88 precision value. From this it is proved that content based Spark-ITFS provides better result than the other methodologies.

Recall

The Recall value is evaluated according to the retrieval of information at true positive prediction, false negative

$$Recall = \frac{True\ positive}{(True\ positive + False\ negative)}$$

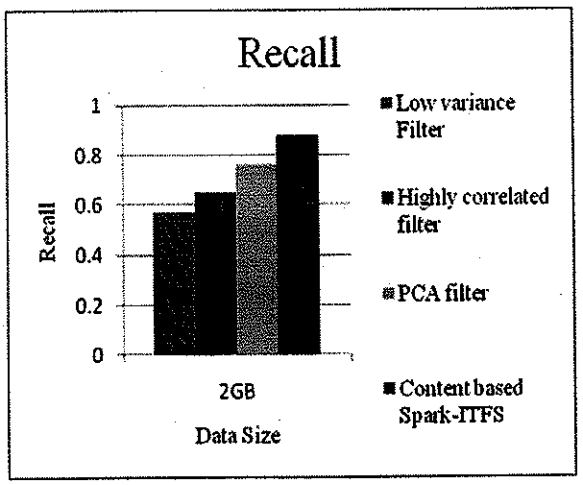

Figure 4 : Recall Comparison

From the figure 4 it is proved that the content partitioning based spark-ITFS can filter efficiently in terms of better recall. From the above graph, it is shown that the Low variance filter provides 0.57 recall value, Highly correlated filter provides 0.65 recall value, PCA filter provides 0.76 recall value, and content based Spark-ITFS provides 0.88 recall value. From this it is proved that content based Spark-ITFS provides better result than the other methodologies.

F-Measure

The F-Measure computes some average of the information retrieval precision and recall metrics

$$F - Measure = \frac{2 * precision.\ recall}{precision + recall}$$

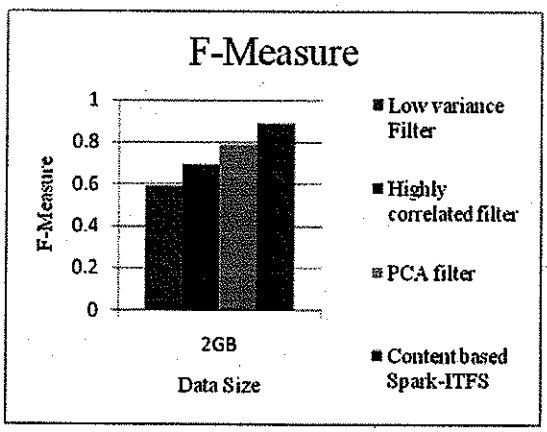

Figure 5 : F-measure

From the figure 5 it is proved that the content partitioning based spark-ITFS can filter efficiently in terms of better f-measure value. From the above graph, it is shown that the Low variance filter provide 0.59 f-measure value, Highly correlated filter provides 0.69 f-measure value, PCA filter provides 0.79 f measure value, and content based Spark-ITFS provides 0.89 f measure value. From this it is proved that content based Spark-ITFS provides better result than the other methodologies.

5. Conclusion

Big data handling management is the most trivial task in the real world environment which plays major role in organization and industries. Filtering is the better approach for mining the useful information from the set of available big data sets to make ease of processing of the real world applications. In this work, content partition based Spark-ITFS approach is introduced which attempts to filter the data in terms of various irrelevancy and redundancy present between them. This proposed research methodology provides better results than the previous filtering approaches in terms of better filtering process. The overall performance evaluation of this work is done in the Hadoop simulation environment which is proved that the proposed content based approach provides better result in terms of improved accuracy, precision, recall and f-measure values.

References

[1] Zhe Wang, Yongji Wang, Hu Wu, *"Tags Meet Ratings: Improving Collaborative Filtering with Tag-Based Neighborhood Method"*, ACM transaction, 2010

[2] Ning Zhou, William K. Cheung, Guoping Qiu, and Xiangyang Xue, *"A Hybrid Probabilistic Model for Unified Collaborative and Content-Based Image Tagging"*, IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 33, No. 7, July 2011

[3] Feng Xie, Zhen Chen, Hongfeng Xu, Xiwei Feng, and Qi Hou, *"TST: Threshold Based Similarity Transitivity Method in Collaborative Filtering with Cloud Computing"*, Tsinghua Science And Technology, ISSN - 11007-0214111/1111, pp318-327 Volume 18, Number 3, June 2013

[4] Mohamed Amine Chatti, Simona Dakova, Hendrik Thu" s, and Ulrik Schroeder, *"Tag-Based Collaborative Filtering Recommendation in Personal Learning Environments"*, IEEE Transactions On Learning Technologies, Vol. 6, No. 4, October-December 2013.

[5] Rong Hu, Wanchun Dou, And Jianxun Liu, *"ClubCF: A Clustering-Based Collaborative Filtering Approach for Big Data Application"*, IEEE Transactions On Emerging Topics In Computing, 2014

[6] Holly Alexander, Ibrahim Khalil, Conor Cameron, Zahir Tari, and Albert Zomaya, *"Cooperative Web Caching Using Dynamic Interest-Tagged Filtered Bloom Filters"*, IEEE Transactions On Parallel And Distributed Systems, Vol. 26, No. 11, November 2015

[7] Juan J. Pomárico-Franquiz, Moises Granados-Cruz, and Yuriy S. Shmaliy, *"Self-Localization Over RFID Tag Grid Excess Channels Using Extended Filtering Techniques"*, IEEE Journal Of Selected Topics In Signal Processing, Vol. 9, No. 2, March 2015.

[8] Shanshan Yao, Yunsheng Wang, and Baoning Niu, *"An Efficient Cascaded Filtering Retrieval Method for Big Audio Data"*, IEEE Transactions On Multimedia, Vol. 17, No. 9, September 2015.

Authors' Biography

Karthick N is a full time research scholar, Department of Computer Science,

Karpagam university. He has attended various national and international conferences. He has published 3 papers in various journals. His area of interest is data mining, big data.

Dr. X. Agnes Kalarani is currently working as Associate Professor, Department of computer Applications, Karpagam University. She received her Ph.D in 2011. She has published 17 publications and attended various national and international conferences. Her area of interest is networking.