Analysis and Performance Evaluation of Cosine Neighbourhood Recommender System

Kola Periyasamy[1], Jayadharini Jaiganesh[1], Kanchan Ponnambalam[1], Jeevitha Rajasekar[1], and Kannan Arputharaj[2]

[1]Department of Information Technology, Anna University, India.

[2]Department of Information Science and Technology, Anna University, India.

Abstract: *Growth of technology and innovation leads to large and complex data which is coined as Bigdata. As the quantity of information increases, it becomes more difficult to store and process data. The greater problem is finding right data from these enormous data. These data are processed to extract the required data and recommend high quality data to the user. Recommender system analyses user preference to recommend items to user. Problem arises when Bigdata is to be processed for Recommender system. Several technologies are available with which big data can be processed and analyzed. Hadoop is a framework which supports manipulation of large and varied data. In this paper, a novel approach Cosine Neighbourhood Similarity measure is proposed to calculate rating for items and to recommend items to user and the performance of the recommender system is evaluated under different evaluator which shows the proposed Similarity measure is more accurate and reliable.*

Keywords: *Big Data, Recommender System, Cosine Neighbourhood Similarity, Recommender Evaluator.*

Received April 28, 2014; accepted June 12, 2014

1. Introduction

The surprising growth in volumes of data has badly affected today's business. The online users create content like blog posts, tweets, social networking site interactions and photos. And the servers continuously log messages about what online users are doing. The online data comes from the posts on the social media sites like facebook and twitter, YouTube video, cell phone conversation records etc. This data is called Big Data.

Big Data [2, 7, 23] concept means a dataset which continues to grow so much that it becomes difficult to manage it using existing database management concepts and tools. The difficulty can be related to data capture, storage, search, sharing, analytics [14] and visualization etc., In present day Recommender System [24], Recommendation is either memory based or model based. In Model-based Recommendation approach, the system develops a model for the given items rated by users. It is trained with the given data and processed to find patterns in the given items based on user's preference. Using the generated model, ratings are calculated for un-rated items and the items with ratings above a threshold value is recommended for each user.

In memory-based Recommendation Approach [1] (Content based filtering), items are recommended for each user based on inter-user similarity. It analyses entire dataset to find users similar to a given user i.e., it finds a set of users whose preference matches with the target user. Such users are called neighbours. Several algorithms are applied to calculate rating for non-rated items based on the neighbour's preference. Rating is calculated for items that are not rated by the users and items are recommended by the ratings. Several similarity measures are applied to calculate the rating such as Cosine similarity, Pearson correlation coefficient, city-block, Count-Based measure etc.

2. Related Work

Kim *et al*. in their paper [16] proposed a theory according to which the evolution of smart phones and Social Network Services (SNS) leads to the big data era. Twitter data has been collected, stored and analyzed in a multi-dimensional fashion on top of Hadoop platform in order to find out what kind of factors can affect the customer preference for the smart phones. About 600,000 Twitter data [21] has been collected for one month and the analysis result shows the most popular Smartphone, the most interesting attributes in the smart phones, and the maker the customers most interested in.

Han *et al*. in their paper [12] proposed a big data model for recommender systems using social network data. The model incorporates factors related to social networks and can be applied to information recommendation with respect to various social behaviours that can increase the reliability of the recommended information.

The Big Data model [15] has the flexibility to be expanded to incorporate more sophisticated additional factors if needed. The experimental results using it in information recommendation and using map-reduce to process it show that it is a feasible model to be used for information recommendation. Anderson [3] ,in their paper, proposed an adaptive and Learning based Recommender system.

One of the important personalization technologies in the information recommendation [10] system is collaborative filtering. Collaborative Filtering (CF) [4] is the process of filtering or evaluating items through the opinions of other people. CF technology brings together the opinions of large interconnected communities on the web [17], supporting filtering of substantial quantities of data. There exists many approaches to achieve recommendations like basic techniques of collaborative filtering and content based approach. These approaches can be done individually or combined depending on the type of recommendations needed by individuals.

Gunawardana and Shani in their paper [11], used RMSE as a metric for scoring an algorithm. It is stated that larger error are penalized severely by RMSE than other metrics. Prediction task can also be performed by RMSE values as they measure inaccuracies on all ratings both positive and negative. Hernandez and Elena in their paper [8], assumed that a "successful recommendation" is equivalent to "the usefulness of the recommended object is close to the user's real preferences" and classified Mean Absolute Error (MAE) as Rating prediction which measures the capacity of the recommender system to predict the preference a user will give to an item.

Vozalis and Margaritis in their paper [22] discuss an evaluation metric recall-precision to evaluate the value of the recommended results. They divide their dataset into two disjoint sets, the training sets and test sets. Cremonesi *et al.* in their paper [6], uses accuracy metrics precision and recall to determine the performance of MovieLens dataset and Netflix dataset. Boyd *et al.* in their paper [5] evaluated the recommender system's performance using Precision-Recall (PR) curves. PR curves are used when the distribution of data is skewed.

Cosine similarity is an established similarity measure which produces accurate results. It is a similarity measure between two vectors which measure the cosine angle between them. It is also used in information retrieval and text mining to compare several text documents. The similarity between two items a and b is denoted in Equation 1.

$$c_{a,b} = \frac{\sum_{u\in U}\left(r_{u,a}-\hat{r}_a\right)\left(r_{u,b}-\hat{r}_b\right)}{\sqrt{\sum_{u\in U}\left(r_{u,a}-\hat{r}_a\right)^2}\sqrt{\sum_{u\in U}\left(r_{u,b}-\hat{r}_b\right)^2}} \quad (1)$$

Where $r_{u,a}$ is the rating of user u on item a, $r_{u,b}$ is the rating of user u on item b, $\hat{r}_a$ is the average rating on a-th item and $\hat{r}_b$ is the average rating on b-th item. Pearson Correlation coefficient is measure of similarity between two items which is denoted by Equation 2.

$$w_{a,j} = \frac{\sum_j\left(r_{a,j}-\overline{r_a}\right)\left(r_{k,j}-\overline{r_k}\right)}{\sqrt{\sum_j\left(r_{a,j}-\overline{r_a}\right)^2\sum_j\left(r_{k,j}-\overline{r_k}\right)^2}} \quad (2)$$

Where r_a and r_k are the averages of customer a's ratings and customer k's ratings, respectively. $r_{k,j}$ is customer k's ratings for item j and $r_{a,j}$ is customer a's rating for item j. If customers a and k have a similar rating for an item, $w_{a,k}> 0$. | $w_{a,k}$ | indicates how much customer a tends to agree with customer k on the items that both customers have already rated. If they have opposite ratings for an item, $w_{a,k}< 0$ and | $w_{a,k}$ | indicates how much they tend to disagree on the item that both again have already rated. Hence, if they don't correlate each other, $w_{a,k}=0$. $w_{a,k}$ can be in between -1 and 1. However it has a serious problem when there are very few items rated by the users. This problem occurs when the amount of items become very large reducing the number of items users have rated to a tiny percentage.

The paper is organized as follows: section 2 details about the Related Work. Proposed Cosine Neighbourhood Recommender (CNR) system architecture is given in section 3 describing the modules. Implementation details are discussed in section 4. Performance of the proposed system is analyzed in section 5 and concluded in section 6.

3. Proposed System

The block diagram of the proposed CNR system is illustrated in Figure 1.

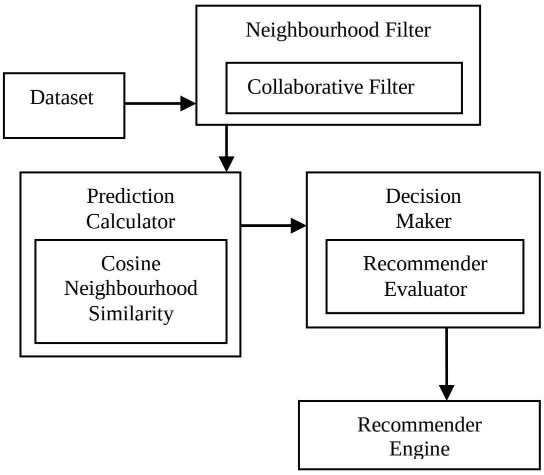


Figure 1. Overall system architecture for cosine neighbourhood recommender system.

This system consists of three major modules. Initially Data from social media is obtained and required data of the users are filtered to get UserId, ItemId and the rating given by users for items. Neighbourhood Filtering module filters those users whose taste matches with the user u. Prediction Calculator module calculates rating for every non-rated item based on inter-user similarity using the proposed Cosine Neighbourhood (CN) similarity.

Decision Evaluator analyses the quality of the rating by recommender evaluation techniques which calculates the evaluation score of the system. The results of item to be recommended to the users are displayed by incorporating Mahout Framework.

3.1. Neighbourhood Filter

Recommender system is a system which processes the available data about the users and the items preferred by the users to generate a list of items which the user will prefer in future. It can be done in three ways- Collaborative filtering, Content-Based filtering and hybrid recommender System. In content-based filtering, the user's history data is analyzed. Items are recommended to users based on the items which they preferred in the past. In collaborative filtering, items are recommended to users based on their similarity to other users. In hybrid approach, both the user's history and inter-user similarity is taken into consideration.

In neighbourhood filter module, top users are filtered from the entire set of users by applying collaborative Filtering. Collaborative Filtering is the mostly commonly used Recommendation Approach. It recommends items to user based on their taste. i.e., Recommendation is based on the user's preference to other items.

Collaborative recommender systems analyze ratings given by users for items. It recognizes similarities between users based on the ratings and calculate ratings for un-rated items based on inter-user comparisons. Figure 2 represents the flow in collaborative filtering. There are two popular approaches in collaborative Filtering: model-based and memory-based filtering approach.

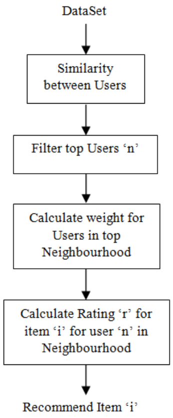

DataSet
Similarity between Users
Filter top Users 'n'
Calculate weight for Users in top Neighbourhood
Calculate Rating 'r' for item 'i' for user 'n' in Neighbourhood
Recommend Item 'i'

Figure 2. Flow diagram for neighbourhood filter and prediction calculator.

3.2. Prediction Calculator Module

Similarity computation between users is a significant step in memory-based Collaborative Filtering Algorithms. For item-based CF algorithms, the basic idea of the similarity computation between items i and item j is first to work on the users who have rated both of these items and then to apply a similarity computation to determine the similarity between the two co-rated items of the users. In this system a novel similarity measure, Cosine Neighbourhood Similarity, is proposed which calculates the rating for non-rated items.

For a user-based CF algorithm, the similarity between the users u and v who have both rated the same items is calculated. There are many different methods to compute similarity or weight between users or items. Weighted Rating is calculated for all non-rated items using Pearson correlation Coefficient.

Cosine Neighbourhood similarity is an established similarity measure which produces accurate results. It is a similarity measure between two users 'u' and 'v' which measure the cosine angle between them for a single item 'a'. The similarity between two users u and v over an item 'a' is denoted in Equation 3.

$$c_{u,v} = \frac{\Sigma_{u\in U}\left(r_{u,a}-\overline{r}_u\right)\left(r_{v,a}-\overline{r}_v\right)}{\sqrt{\Sigma_{u\in U}\left(r_{u,\mathrm{a}}-\hat{r}_u\right)^2}\sqrt{\Sigma_{v\in U}\left(r_{v,a}-\hat{r}_v\right)^2}} \qquad (3)$$

Where $r_{u,a}$ is the rating of user u on item a, $r_{u,b}$ is the rating of user u on item b, $\hat{r}_u$ is the average rating given by user u on all items and $\hat{r}_v$ is the average rating given by user v on all items.

The flow for Neighbourhood Filter and Prediction Calculator module is depicted in Figure 2. Top Neighbors are filtered and rating is calculated for items that are not rated by the users u in the Neighbourhood G. Figure 3 represents the activity diagram to calculate rating for users in Neighbourhood.

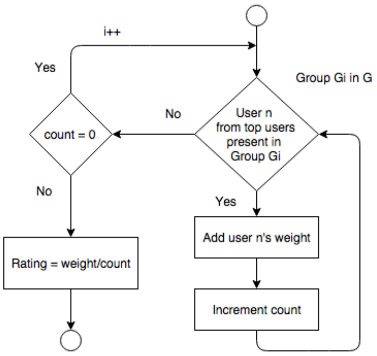


Figure 3. Activity diagram for rating calculation.

3.3. Decision Maker

In decision maker module, the above proposed recommender System is evaluated for its performance and accuracy. Recommender system uses many methods to evaluate the quality of the recommendation system. There are two major accuracy metrics to evaluate a recommender system. They are Decision support metrics and Statistical support accuracy metrics. In decision Support accuracy metrics, the ability of the system the system to help the user to select a high quality items from the set of all non-rated items is evaluated.

In Statistical accuracy metrics, Performance of Recommender system is evaluated based on the accuracy of the recommender data. i.e., how far they match with the original data.

Statistical Accuracy Metrics evaluates the quality of the recommender system. It calculates evaluation score using which it evaluates the quality of recommended items. Few rated items are considered unrated and the recommender engine is made to generate ratings for those items. The difference between the actual rating and the generated rating is the Mean Absolute Error which determines the quality of the recommender system. Two methods are applied in this paper to evaluate the accuracy. They are average absolute difference evaluator and RMS recommender evaluator.

In average absolute difference evaluator, Mean Absolute Error (MAE) is calculated which is the evaluation score. It measure how close the calculated value is against the original ratings. It is given by Equation 4.

$$MAE = \frac{1}{n}\Sigma_{i=1}^{n}\left| f_i - y_i \right| \tag{4}$$

Where f_i is the original rating given by the user for an item, y_i is the rating generated by the recommender system, n is the number of users and i is iterator. The lesser the value of MAE, the greater the quality of the Recommender system.

In RMSRecommenderEvaluator, Root Mean Square Error (RMSE) is calculated which serves as the evaluation score. This is the square root of the average of this difference, squared. It is given by Equation 5.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(f_i - y_i \right)^2}{n}} \tag{5}$$

Where f_i is the original rating given by the user for an item, y_i is the rating generated by the recommender system, n is the number of users and *i* is iterator.

Precision and Recall are a set of accuracy metrics used to evaluate the value of the recommender system's results. Precision is defined as the ability of a system to retrieve all the relevant data or items. It is denoted by Equation 6.

$$precision = \frac{\left|\{r_e\} \cap \{r_i\}\right|}{\left|r_i\right|} \tag{6}$$

Where r_e is relevant documents and r_i is retrieved documents. Recall is defined the ability of the recommender system to retrieve only the relevant items. It is given by Equation 7.

$$recall = \frac{\left|\{r_e\} \cap \{r_i\}\right|}{\left|r_e\right|} \tag{7}$$

Where r_e is relevant documents and r_i is retrieved documents.

4. Implementation

Dataset -MovieLens Datasets- [9] consists of columns of UserId, ItemId and the rating given by user for items. This data set consists of 100,000 ratings (1-5) from 943 users on1682 movies. Each user has rated at least 20 movies which are represented as items.

Since dataset is large and complex, Hadoop framework [13] is used to process data as it provides scalability and flexibility. Collaborative Filtering approach is used to recommend items to user. It considers inter-user similarity to calculate rating for non-rated items and recommend the items with higher ratings. For user u, Rating for every non-rated item is calculated by considering the preference given by those users who have rated the items already rated by user u. For each user, rating for every non-rated item is calculated using two approaches- Pearson correlation coefficient and Cosine similarity. Rating is calculated for all non-rated items. The rating value ranges from 0-1 where 0 indicates that the item will not preferred by user u. Rating 1 indicates highest priority. Rating is calculated using both Pearson Correlation and Cosine vector similarity method. For every non-rated item, the calculated rating is listed and items with the top ratings are considered for recommendation.

MapReduce [19, 20] is a programming model with which large datasets can be processed in parallel and distributed manner. MapReduce paradigm is used to calculate rating for items and to recommend the items to each user. It is composed of two major functions- Map () and Reduce ().The master node of map () takes input and it is divided into smaller tasks which are then assigned to several child nodes or workers nodes. Input is given in the form of <Key, value> pairs to the map() function.

The output of mapper is of a list of values for a key. Here the <UserId, ItemId, rating> is given to the master nodes. It manipulates the data and calculates rating for non-rated items for each user u based on inter-user similarity. Rating is calculated using two similarity metrics- Pearson coefficient and cosine vector similarity.

Mahout [18] is also incorporated for recommending items to users. It is a machine learning framework which acts as a programming interface built on the top of Hadoop MapReduce. Mahout constructs a similarity matrix based on the user preference and the calculated Ratings. It uses series of Mappers and Reducers to make the recommendations.

5. Performance Analysis

Recommender Evaluator evaluates the performance of the recommender system in terms of both accuracy and quality. Statistical accuracy metrics [11] calculates an evaluation score for the recommender system. Two approaches are used to calculate the evaluation score. AverageAbsoluteDifference Evaluator calculates MAE in which a set of rated items are considered non-rated and they are subjected to the recommender system. It calculates ratings for those items.

The average of the difference between original rating and the calculated rating is the mean absolute error. MAE is calculated under varying conditions of user preference i.e, percentage of user preference is varied. Other method used to evaluate the score is RMSRecommenderEvaluator. This evaluator calculates the RootMeanSquareError which is the square root of the difference between original rating and the calculated rating. The evaluation score thus calculated using two methods under different percentage if user preference is tabulated.

The output of mapper is of a list of values for a key. Here the <UserId, ItemId, rating> is given to the master nodes. It manipulates the data and calculates rating for non-rated items for each user u based on inter-user similarity. Rating is calculated using two similarity metrics- Pearson coefficient and cosine vector similarity. The output of mapper is <UserId, List (ItemId, Rating)> for all non-rated items.

This is fed to the reduce () function which filters the top non-rated items based on the calculated ratings. Figures 4 and 5 represent MAE values and RMSE values for CNS under different percentage of User Preference respectively.

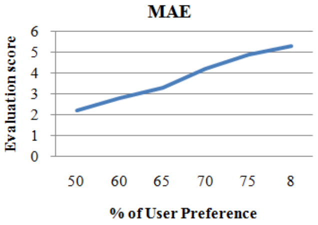


Figure 4. Evaluation score (MAE) for CNS.

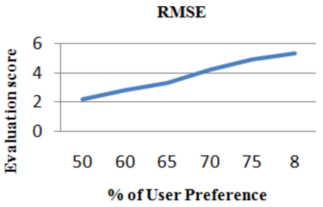


Figure 5. Evaluation score (RMSE) for CNS.

In the graphs, evaluation scores are plotted against the percentage of user preference. Evaluation score is calculated and compared for the recommender system under different similarity coefficient.

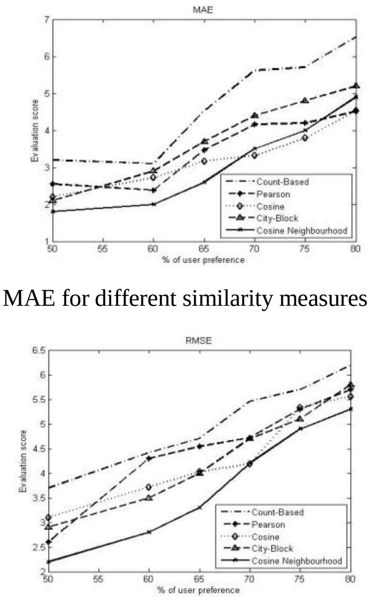
MAE for different similarity coefficient is plotted in a graph with an evaluation score in y-axis and the percentage of user preference in x-axis is shown in Figure 6-a. Similarly, Figure 6-b shows RMSE for different similarity measures with an evaluation score in y-axis and the percentage of user preference in x-axis. In both these graphs, the evaluation score for Cosine Neighbourhood Similarity is comparatively greater than other similarity measures. This illustrates that the CNS is an accurate and reliable similarity measure. Evaluation scores in terms of MAE and RMSE are tabulated for different similarity coefficients under varying percentage of user preference in Tables 1 and 2 respectively.

Table 1. Comparison of Evaluation score calculated using MAE under different similarity coefficients.

Similarity Measure % of User Preference	Count-Based	Pearson	Cosine	City Block	Cosine Neighbourhood
80	6.82	4.51	4.53	5.2	4.9
75	6.71	4.2	3.79	4.8	4.0
70	5.62	4.16	3.32	4.4	3.5
65	4.53	3.47	3.17	3.7	2.6
60	4.1	2.38	2.72	2.9	2.0
50	3.2	2.05	2.20	2.1	1.8

Table 2. Evaluation score calculated using RMSE under different similarity coefficients.

Similarity Measure % of User Preference	Count-Based	Pearson	Cosine	City Block	Cosine Neighbourhood
80	6.2	5.7	5.56	5.8	5.3
75	5.7	5.3	5.33	5.1	4.9
70	5.46	4.72	4.19	4.7	4.2
65	4.31	4.54	4.03	4	3.3
60	4.42	4.5	3.72	3.5	2.8
50	3.7	1.6	3.1	2.9	2.2



a. MAE for different similarity measures .

b. RMSE for different similarity measures.

Figure 6. MAE for different similarity measures and RMSE for different similarity measures.

Error rate, in terms of MAE and RMSE, is calculated for recommendations made using all similarity measures. It is proved that Recommendations made using Cosine Neighbourhood Similarity is 5.67% more accurate than Cosine Similarity under MAE and 3.7% more accurate than Cosine Similarity under RMSE. CNR System generates recommendations which are 4.6% more accurate and reliable than Cosine Similarity. The Accuracy Percentage of CNS against some similarity measures is tabulated in Table 3.

Table 3. Accuracy percentage for cosine neighbourhood similarity.

Evaluation Score / Similarity Measure	Increase in Accuracy (%)	
	MAE	**RMSE**
Cosine Neighbourhood	5.67	3.7
Cosine	4.5	3.4
City-Block	4.1	3.33
Pearson	2.75	5.06
Count-Based	3.6	6.09

Precision and Recall values are calculated for the recommended results for both the Cosine Neighbourhood and Cosine Similarity measures. Recall values are set to 0, 0.1, 0.2 till 1 and the PR curves are plotted using Precision values against these points. Figure 7 shows the comparison of CNS and Cosine Similarity measures using PR curves.

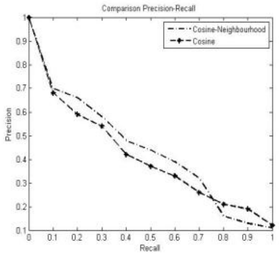


Figure 7. Comparison of CNS and cosine similarity using PR curves.

Precision values of Cosine Neighborhood Similarity measure are comparatively greater than that of Cosine Similarity measure. High precision means that an algorithm returned substantially more relevant results than irrelevant. This shows that CNS measure retrieves more accurate data to recommend the users. Thus it is shown that the proposed CNS measure has improved performance than Cosine Similarity measure.

6. Conclusions

In this paper, a novel Cosine Neighbourhood Similarity measure is proposed which is incorporated in Recommender System. CNS significantly improves the scalability and the quality of the recommendations made to the users. It is more accurate and reliable than other similarity measures since users are filtered using collaborative filtering before calculating the ratings which also fastens the recommendation process. PR curves indicate that Cosine Neighbourhood Similarity retrieves more accurate data than Cosine Similarity. Thus it also improves the quality of recommendation by reducing the error rates and reduces the latency.

References

[1] Abbas A. and Liu J., "Designing an Intelligent Recommender System using Partial Credit Model and Bayesian Rough Set," *The International Arab Journal of Information Technology*, vol. 9, no. 2, pp. 179-187, 2012.

[2] Adomavicius G. and Tuzhilin A., "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, 2005.

[3] Anderson C., "Real Time Data from Big Time Fun: Harnessing the Power of Mobile Technology and Social Media for Better Species Management," *in proceeding of 2012 Oceans*, California, pp.1-5, 2012.

[4] Bell R. and Koren Y., "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights," *in proceeding of Seventh IEEE International Conference on Data Mining (ICDM 2007)*, USA, pp. 43-52, 2007.

[5] Boyd K., Costa V., Davis J. and Page C., "Unachievable Region in Precision-Recall Space and Its Effect on Empirical Evaluation," *in Proceeding of the 29th International Conference on Machine Learning*, Edinburgh, pp. 1-8, 2012.

[6] Cremonesi P., Koren Y., and Turrin R., "Performance of Recommender Algorithms on Top-N Recommendation Tasks," *in Proceeding of RecSys'10 Proceedings of the 4th ACM Conference on Recommender Systems*, New York, pp. 39-46, 2010.

[7] Demchenko Y., Zhiming Z., Grosso P., and Wibisono, A., de Laat, C., "Addressing Big Data challenges for Scientific Data Infrastructure," *in proceeding of IEEE 4th International Conference on Cloud Computing Technology and Science*, Taipei, pp. 614-617, 2012.

[8] Elena G. and Olma F., "Evaluation of recommender systems: A new approach," Expert Systems with Applications, vol. 35, no. 3, pp. 790-804, 2008.

[9] GroupLens, http://grouplens.org/datasets/movielens, Last Visited 2014.

[10] Gubanov M. and Pyayt A., "MEDREADFAST: A Structural Information Retrieval Engine for Big Clinical Text," *in proceeding of IEEE 13th International Conference on Information Reuse and Integration (IRI)*, USA, pp. 8-10, 2012.

[11] Gunawardana A., Shani G., "Survey of Accuracy Evaluation Metrics of Recommendation Tasks,"

Journal of Machine Learning Research, vol. 10 pp. 2935-2962, 2009.

[12] Han X., Tian L., Yoon M., and Lee M., "A Big Data Model Supporting Information Recommendation in Social Networks," *in proceeding of Second International Conference on Cloud and Green Computing*, Washington, pp. 1-3, 2012.

[13] Humbetov S., "Data-intensive Computing with Map-Reduce and Hadoop," *in procedding of 6th International Conference on Application of Information and Communication Technologies*, Georgia, pp. 17-19, 2012.

[14] Ibm, http://www.ibm.com/software/data/infosphere/hadoop/what-is-big-data-analytics.html, Last Visited 2014.

[15] Kaisler S., Armour F., Espinosa J., and Money W., "Big Data: Issues and Challenges Moving Forward," *in procedding of 46th Hawaii International Conference on System Sciences*, Wailea, pp. 7-10, 2013.

[16] Kim J., Yang M., Hwang Y., Jeon S., Kim K., Jung I., Choi C., Cho W., and Na J., "Customer Preference Analysis Based on SNS Data," *in procedding of Second International Conference on Cloud and Green Computing*, Washington, pp. 609-613, 2012.

[17] Li N., Zhang N., and Das S., "Preserving Relation Privacy in Online Social Network Data," *Internet Computing IEEE* , vol. 15, no. 3, pp.35-42, 2011.

[18] Mahout, http://mahout.apache.org/users/recommender/recommender-documentation.html, Last visited 2014.

[19] Martha V., Zhao W., and Xu X., "h-MapReduce: A Framework for Workload Balancing in MapReduce," *in procedding of IEEE 27th International Conference on Advanced Information Networking and Applications*, Barcelona, pp. 637-644, 2013.

[20] Palit I. and Reddy C., "Scalable and Parallel Boosting with MapReduce," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 10, pp. 1904-1916, 2012.

[21] Smith M., Szongott C., Henne B., Von V., "Big Data Privacy Issues in Public Social Media," *in procedding of 6th IEEE International Conference on Digital Ecosystems Technologies*, Campione d'Italia, pp.1-6, 2012.

[22] Vozalis E. and Margaritis E., "Analysis of Recommender Systems' Algorithms," *in Proceeding of the 6th Hellenic European Conference on Computer Mathematics and its Applications*, Athens, pp. 1-14, 2003.

[23] Wikipedia, http://en.wikipedia.org/wiki/Big_data, Last Visited 2014.

[24] Wikipedia, http://en.wikipedia.org/wiki/Recommender_system, Last Visited 2014.

Kola Sujatha Periyasamy is a Senior Assistant Professor in Madras Institute of Technology of Anna University, India. She received her M.C.A. in Computer Applications in 1999, M.E in Computer Science and Engineering in 2003 and her Ph.D degree in Computer Science and Engineering in 2013 from College of Engineering, Guindy, Anna University, India. She has 11 years teaching experience in the branch of Information Technology. Her current research focuses on Data Mining and Big Data Analytics.

Jayadharini Jaiganesh completed her Bachelor Degree in Information Technology in Madras Institute of Technology, Anna University, India. Her areas of interests are data mining and image processing. She has published a paper IEEE Conference. She is pursuing her research in the area of data mining.

Kanchan Kumar Ponnambalam completed his Bachelor Degree in Information Technology in Madras Institute of Technology, Anna University, India. He is well versed in Photoshop. His areas of interests are data mining and Operating Systems.

Jeevitha Rajasekar completed her Bachelor Degree in Information Technology in Madras Institute of Technology, Anna University, India. Her areas of interests are data mining and soft computing.

Kannan Arputharaj is a Professor and Head of the Department of Information Science and Technology, College of Engineering, Anna University, India. He received his B.Sc. degree in Mathematics from Madurai Kamaraj University in 1979, his Master degree in Mathematics from Annamalai University in 1986, his M.E degree in Computer Science and Engineering from College of Engineering, Guindy, Anna University, India in 1991 and his Doctorate in Intelligent Temporal Databases from the Faculty of Electrical Engineering, Anna University, India in 2000. He worked as a Programmer in Bhabha Atomic Research Centre (BARC), Mumbai, India from 1981 to 1989. He has published numerous papers in various International journals and conferences. He has 27 years of teaching experience. His area of research includes Database Management System, Artificial Intelligence, Big Data Analytics, Data Mining and Software Engineering.