A Cuisine Based Recommender System Using k-NN And Mapreduce Approach

Sadhana Kodali, Madhavi Dabbiru, B Thirumala Rao

Abstract**:** ***In the present days, life can be made smarter, including the food we eat by taking an option from the restaurant recommender systems. In this paper the authors proposed a restaurant recommender system based on the search of user cuisine. The top-k restaurants are identified along with the ratings of the restaurants recommended. The recommendations are retrieved based on the preference of the user cuisines which is an important category which inherently defines the other features and these features are considered to provide a good service which is the novelty of this paper. Providing recommendations based on user cuisines is the complexity of the problem. The well known k-Nearest Neighbor algorithm is implemented with the MapReduce paradigm which can quickly process huge amounts of data. Its performance is tested on benchmarked data set and the results are found to be successful.***

Index Terms**:** ***Restaurant Recommender System, Nearest Neighbor approach, MapReduce, Cuisine based search.***

I. INTRODUCTION

In this world of competition it is essential to be healthy and choose proper food to eat by making use of the improved resources that we have around us. The various search engines provide a set of restaurants that are suitable to us but they do not focus on food specific search. In this paper we propose a restaurant recommender system where the user can get the restaurants based on his taste and he can also have a list of restaurants that provide the food including the ratings of the restaurants which provide a better option for the user to choose. We classify the restaurants based on user specific food by making use of the well known kNN algorithm. This also helps to a group of users who have the same food habits to get good recommendations based on the similarity. This may be helpful to have good, happy and tasty get together where all the group of people have the same savour. With the use of internet and promotion in websites many restaurants data are available which has a great need of good restaurant recommendation system that filters and promotes good suggestion for tasty food. The proposed approach is an application of Machine learning based classification to classify the restaurant.

The food- restaurant-customer form a kind of semantic network where multiple paths can be used to traverse different semantic relationships like: identifying food of same or different type, customers of same age, taste, restaurants of same or different ratings which are nothing but various Meta paths in the Information Network. Information Networks can be classified as Homogeneous Information Networks where objects interact with similar kind of links and Heterogeneous Information Networks[1] where different kind of objects communicate with various semantically interconnecting links. A heterogeneous network can contain a homogeneous network within it. If the homogeneous network formed between food and person is considered the meaningful relationship established is “Person eats Food”. From Figure 1 we can observe that a person may be male, female or children. Food can be of any kind burgers,pizzas,fries etc.But the meaningful interaction among person and food is “eats” which is a link between these objects.

Fig 1: Homogeneous Network formed between person and food.

Figure 2 depicts a heterogeneous network formed between customer-food-restaurant. Different semantic Meta-paths can be formed between the objects shown in figure-2

Example Meta-path:

Customer—visits—restaurant—has a type of—food

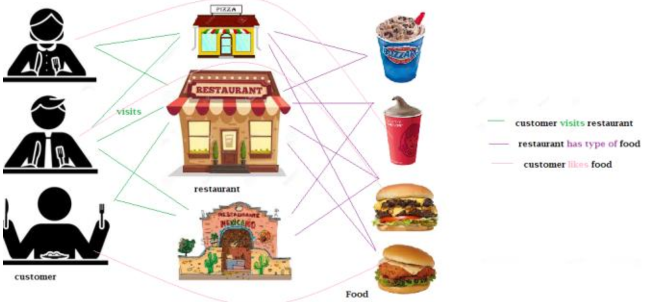


Fig-2: customer-restaurant-Food Network

Revised Manuscript Received on May 10, 2019

Sadhana Kodali, Ph.D Scholar,Department of CSE, Koneru Lakshmaiah Education Foundation,Vaddeswaram, Guntur Dt, AndhraPradesh, India.

Madhavi Dabbiru , Professor&Head , Department of Computer Science Engineering, Dr.L.Bullayya College of Engineering for Women,Visakhapatnam, AndhraPradesh, India.

Thirumala Rao B, Professor,Department of CSE, Koneru Lakshmaiah Education Foundation,Vaddeswaram, Guntur Dt, AndhraPradesh, India.

In order to predict the semantic relationship between the objects in the Information Networks, it is significant to compute the similarity between the objects. Many Similarity algorithms are proposed to compute the similarity between objects present in Information Networks.

With the increase of data automation and the World Wide Web we have heterogeneous Information Networks identified in every field like social media, medical databases data, paper-author network etc. Also with heap of information flow in the network a number of traditional Data Mining techniques like clustering, classification, link prediction etc. are suitable and can be applied on the data present in the Information network. In the past few years Heterogeneous Information Networks has a wide spread applications like finding friends on social networks, link prediction, recommendation of products, food and friends. Apart from applying the Data Mining techniques, similarity measure can also be used to compute the similarity between the objects present in the Information Network. Similarity measure is a statistical measure used in DataMining to compute the likeliness between the objects which communicate with each other. The key factor in the recommender systems is to compute the similarity to compare two items that are closer to each other. In this paper the authors proposed restaurant recommender system based on the user cuisine and also provide the related restaurants which are suitable to the cuisine of the customer, the price of the food in a sorted order and also the distance from the current location of the customer. The paper is organized as follows: In Section 2 the related work is discussed, Section 3 focuses on the Methodology adopted, Section 4 showcase experimental results, Section 5 concludes with future scope and research perspective.

II. Related Work

Recommender systems are more prominent with the increase in usage of web and work on the machine learning principle of learning about the Neighbors for better solution. Recommender systems are classified as collaborative filtering models which are also called as Neighborhood models [2], the second approach is content based filtering also called as knowledge based models and the combination of the former and the later is the hybrid model. With the growing data it is a good option to make use of the MapReduce paradigm for processing the data parallelly. In [3] the authors proposed the user based collaborative filtering recommender algorithm using Hadoop MapReduce. On the similar lines content based recommender systems is proposed in [4].The methodology proposed in [5] is a novel approach of using Bhattacharya coefficient for collaborative filtering in sparse data. Chenyang Li and Kejing He [6] proposed an item based collaborative filtering using MapReduce. Though there are many MapReduce based recommenders proposed, applying kNN with MapReduce is a new idea for identifying the nearest Neighbors of the nominal attributes. In [7] Moon-hee Park et al proposed a restaurant recommender systems based on decision made by group of people in mobile environment using Bayesian Networks to model the preference of the users.

Similarity measure is used to measure the likeness between the objects and their interaction. Many similarity algorithms are proposed to analyze the Information Networks. The similarity algorithms are useful for the prediction of links in the heterogeneous information networks and also to detect duplicate web documents. SimRank [8] is a similarity measure proposed based on random surfers' model. It is mainly used for ranking the web documents. PRank [9] is another similarity algorithm which uses unified framework for computing the similarity. PathSim [10] is a similarity measure which uses a commuting matrix. HeteSim [11] makes use of transition probability matrix.

III. Methodology

In this section we discuss about various terminologies like the Meta-Paths which are formed between the heterogeneous objects. Next we also discuss the algorithm for calculating the similarity between the heterogeneous objects using PathSim measure. A Meta-path is a semantic path that exists between the objects. For example for this food chain network which can be treated as a heterogeneous network we consider three different kinds of objects like user, restaurant and cuisine. Though these are of different kind, now they communicate with each other by the following relationships: uservisitsrestaurant, restaurantofferscuisine, cuisineliked byuser

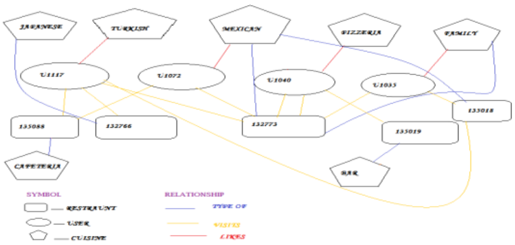


Fig-3: Meta-Path between user-restaurant-cuisine

We have considered the food data set and tried to retrieve the similarity scores for each user, the places they visit and the food they eat. There are many approaches to calculate the similarity but we have chosen the PathSim similarity score as it exhibits properties like [i] symmetric property [ii]self maximum property [iii] Balance of visibility. The PathSim is a meta-path based approach to compute the similarity. According to PathSim, an adjacency matrix must be constructed between these objects to find how one object is similar to another. Instead of an adjacency matrix we build a commuting matrix which is proposed in [12].But any Information Network is huge of its kind as there are number of objects that interact with each other.

When we need to deal with huge datasets which forms the BigData we can make use of the MapReduce strategy which takes the <key,value> pairs as input and produces the output as <key,value> pairs. HDFS (Hadoop Distributed File System) works with Map Reduce to divide the data for parallel processing. The Input files are split into input format, Splits the files into tasks, provides a place for RecordReader Object. The input format defines the list of tasks that forms the Map phase. The Record Reader loads the data and converts into <Key, Value> pairs. The Mappers with the <key, Value> pairs sends these to the reducers. The process of moving the Map output to the Reducers is known as Shuffling. Partitions are the inputs to the Reduce task. Partitioners in Reduce phase identify which are the key value pairs that are stored and reduced. An instance of reducer is created for each Reduce task to create an output format. The Output format governs the way objects are written to the files in HDFS. The Figure 4 represents the Map Reduce procedure.

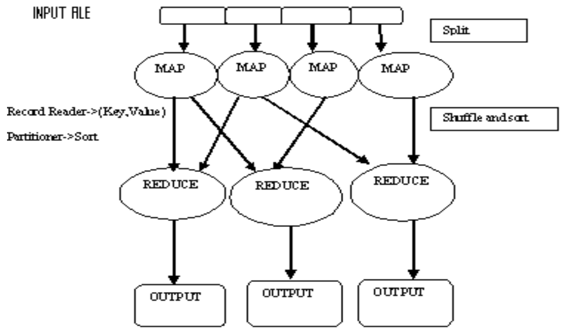
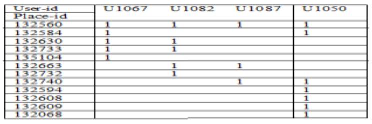


Fig4 Map Reduce Phases

To proceed with, the overall ratings file is given as an input to get the unique location id as the key which is visited by the set of users as value. By considering each row a new Meta-path can be constructed by using the matrix. From the first line of the MapReduce output, it is understood that there are four users (U1067,U1082,U1087,U1050) who visit the place id 132560.By traversing all the other places visited by these four users we get an commuting matrix between the users and places is depicted in figure-4.



User-id Place-id	U1067	U1082	U1087	U1050
132560	1	1	1	1
132584	1			1
132630	1	1		
132733	1	1		
135104	1			
132663		1	1	
132732		1		
132740			1	1
132594				1
132608				1
132609				1
132068				1

Fig 5: Adjacency Matrix for user and place.

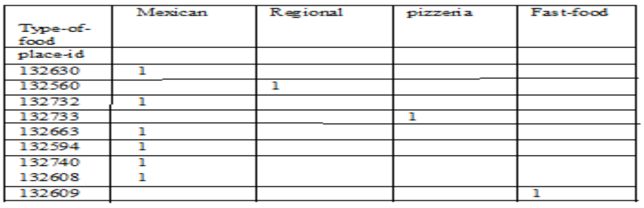
Figure 5 depicts the commuting matrix between the user-id and the restaurant-id. A '1' represents the user visits the place and an empty gap represents that he did not visit the place. From this commuting matrix we can compute the similarity between two users using the formula

$$S(x:y)=2(P_{x\text{->}y} : P_{x\text{->}y} \in P) / \mid (P_{x\text{->}x} : P_{x\text{->}x} \in P)+ \mid(P_{y\text{->}y} : P_{y\text{->}y} \in P) \mid \text{..Eqn(1)}$$

This can be computed as follows for the above example:
Similarity (U1067, U1082) =A/B =0.666
A = 2[(1X1)+(1X0)+(1X1)+(1X1)+(1X0)+(0X1)+(0X1)]
B = [(1+1+1+1+1) + (1+1+1+1+1)]

Similarity (U1067, U1087) =0.25
Similarity (U1067, U1050) =0.166
Similarity (U1082, U1087) =0.25
Similarity (U1087, U1050) =0.4
Similarity (U1082, U1050) =0.0571

In the same way we can also compute the similarity between the restaurant and the cuisine it provides. After doing this we get a Meta-Path between the user-restaurant-cuisine. The figure 6 depicts the commuting matrix formed between the restaurant id and the cuisine offered in the restaurant.



Type-of-food place-id	Mexican	Regional	pizzeria	Fast-food
132630	1			
132560		1		
132732	1			
132733			1	
132663	1			
132594	1			
132740	1			
132608	1			
132609				1

Fig 6: Commuting Matrix between cuisine and restaurant-id.

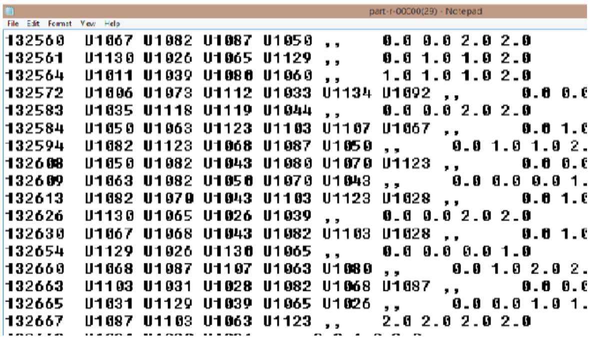
Once we obtain the commuting matrix we can calculate the similarity between the heterogeneous objects using the Algorithm-1 as follows:

Algorithm to find the similarity of Meta-paths between the heterogeneous objects

```
Step 1:Input X
Step 2: Construct the commuting matrix.
Step 3: for all
        Apply the PathSim similarity measure using Eqn 1.
        $S(x:y)=2(P_{x\text{->}y} : P_{x\text{->}y} \in P) / \mid (P_{x\text{->}x} : P_{x\text{->}x} \in P)+ \mid(P_{y\text{->}y} : P_{y\text{->}y} \in P)\mid$
```

Algorithm-1: To compute the similarity between the heterogeneous objects.

The Figure-7 represents the Input file X on which the first MapReduce task is run to get the list of unique restaurant ids, the file will also provide you the list of user ids and the ratings given by them.



```
132560   U1067 U1082 U1087 U1050 ,,      0.0 0.0 2.0 2.0
132561   U1130 U1026 U1065 U1129 ,,      0.0 1.0 1.0 2.0
132564   U1011 U1039 U1080 U1060 ,,      1.0 1.0 1.0 2.0
132572   U1006 U1073 U1112 U1033 U1134 U1092 ,,       0.0 0.0
132583   U1035 U1118 U1119 U1044 ,,      0.0 0.0 2.0 2.0
132584   U1050 U1063 U1123 U1103 U1107 U1067 ,,       0.0 1.0
132594   U1082 U1123 U1068 U1087 U1050 ,,      0.0 1.0 1.0 2.
132608   U1050 U1082 U1043 U1080 U1070 U1123 ,,       0.0 0.0
132609   U1063 U1082 U1050 U1070 U1043 ,,      0.0 0.0 0.0 1.
132613   U1082 U1070 U1043 U1103 U1123 U1028 ,,       0.0 1.0
132626   U1130 U1065 U1026 U1039 ,,      0.0 0.0 2.0 2.0
132630   U1067 U1068 U1043 U1082 U1103 U1028 ,,       0.0 1.0
132654   U1129 U1026 U1130 U1065 ,,      0.0 0.0 0.0 1.0
132660   U1068 U1087 U1107 U1063 U1080 ,,      0.0 1.0 2.0 2.
132663   U1103 U1031 U1028 U1082 U1068 U1087 ,,       0.0 0.0
132665   U1031 U1129 U1039 U1065 U1026 ,,      0.0 0.0 1.0 1.
132667   U1087 U1103 U1063 U1123 ,,      2.0 2.0 2.0 2.0
```

Fig-7: Snapshot of Input File obtained after running the MapReduce

The algorithm to write the first phase of MapReduce before data pre-processing is as follows:

MapReduce paradigm is a parallel approach for processing huge data in the distributed platform. We use Hadoop Distributed File System for the storage of the dataset and implement the MapReduce which takes the <key,value > pair as input and gives the <key,value> pair as output.

Every record id in the document is given as a key and each line is given as input value. After the MapReduce task is completed we get the place id as key and the list of users and their corresponding ratings as value.

```
Algorithm: MapReduce to retrieve the list of restaurant ids and userids who visit the restaurant.
Step 1:Input Ratings_final.csv file
        L is the LongWritable address of each line, Line of text which contains userid and restaurant id.
Step 2: The Map() function
 for all(L,Line of Text)
split(",") and tokenize each restaurant id assign it to data[0]
                              each user_id assign it to data[1]

Step 3: context.write(data[0]+data[1],ratings)
Step 4:end for.
Step 5:The Reduce() function
Step 6:for all<list of values>
Step 7:Use Comparator to sort the ratings
Step 8:end for
Step 9:context.write(<key,Value>);
```

Algorithm-2: To prepare the input file from raw dataset.

The main aim of the authors is to classify the restaurants using the well known algorithm kNN with the MapReduce flavour. Using the kNN we identify the nearest Neighbor to classify the restaurants based on which cuisine most of its users are likely to eat in that restaurant.

k-NN algorithm is a distance based classification algorithm which assigns a class label of an unknown case to its nearest Neighbor. Distance metrics like Euclidean, cosine and Manhattan distances can be used for each Neighbor if it is a continuous attribute. The Euclidean distance is used in the case of continuous variables but with nominal variables we use the Hamming Distance. The cuisine op distance can be stated as dist (hamming) =$|obj_i\text{-}obj_j|$.-Eqn2.

As an example if the user U1067 prefers Mexican dish and U1082 also prefers Mexican then the Hamming Distance between U1067 and U1082 is zero. Else if U1082 prefers Chinese to Mexican it is one. The Algorithm 3 as mentioned below does the classification.

```
Algorithm: MapReduce to classify the list of restaurant ids and based on user cuisine.
Step 1:Pre-processed Input file X
Step 2: The Map() function
         for all(L,Line of Text)
split(",") and tokenize each restraunt id as Text s[0];
                              each user_id as Text s[1];
Assign the user_id to a HashMap
Calculate the Hamming Distance by using if map.containsKey();
                   While(itr.hasnext())
                         int val=map.get(ky);
                         int max=0;
                         String str="";
                        If(val>max)
                            max=val;
                            Str=ky;
Step 3: context.write(s[0],str)
Step 4:end for.
Step 5:The Reduce() function
Step 6:for all<list of values>
Step 7:context.write(<key,Value>);
```

Algorithm-3: To classify the restaurants.

IV. EXPERIMENTAL RESULTS

The experiment is conducted on a single node with 8 GB RAM, 500GB HDD, intel core i3 processor with a speed of 2.86 GHz. The average runtime of all the records in the file turned to be 3.59Milli-seconds. Hadoop is installed in pseudo distributed mode. The input is stored onto HDFS as input file. The input file consists of the restaurant id as the first field with a tab separated space and the user-ids and the cuisine they are interested in using comma separated values.

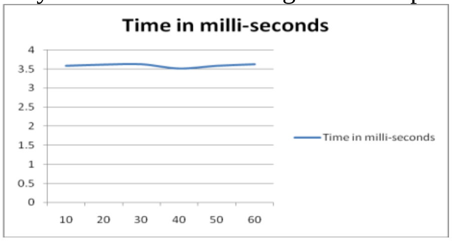


Fig- 8: Graph showing no. of records and run time

4.1 Description of data Set

The data set is taken from the UCI Machine Learning repository from the DataFolder with the URL https://archive.ics.uci.edu/ml/datasets/Restaurant//consume data.The raw data set contains 1161 instances. The first column gives the user_id the second column provides the restaurant_id and three kinds of ratings are provided column three specifies the overall rating column five and six provide the food and service rating respectively. The cuisines offered by the restaurants are 59 but of which we classify the restaurant on the maximum savour of the users who visit the restaurant.

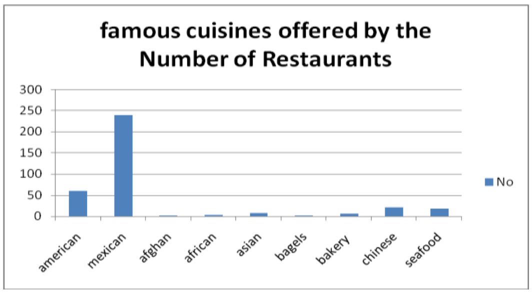


Fig-9: Famous cuisines offered by the number of restaurants

The figure-9 represents the graph in which we can understand that the Mexican is the most famous cuisine offered by 239 restaurants and the classification results also agree with this fact.

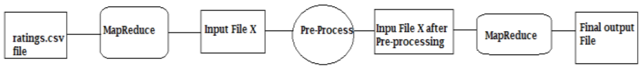
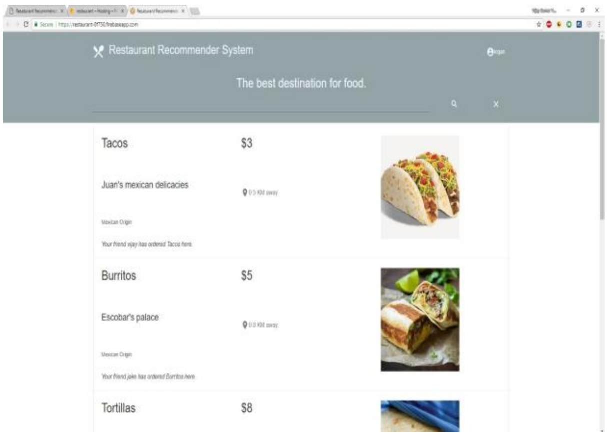


Fig-10: Steps applied to experiment

From the figure-10 the procedure adopted to get the restaurants classified is depicted. The Data pre-processing is done by identifying which user_id will order which cuisine and replaced with that particular cuisine. The empty values are eliminated and the data cleaning is done. The refined and processed file is given as input to the second phase of MapReduce.

The Restaurant Recommender application is developed by the authors and we can view user interface shown in figure 11.



Restaurant Recommender System

The best destination for food.

Tacos $3

Juan's mexican delicacies 0.5 KM away

Mexican Origin

Your friend vijay has ordered Tacos here

Burritos $5

Escobar's palace 0.3 KM away

Mexican Origin

Your friend jake has ordered Burritos here

Tortillas $8

Fig- 11: User Interface of Restaurant recommender system

The predicted top-k recommendations given by the Recommender system are evaluated and shown in figure 11. The classification accuracy is calculated using the formula given below.

$$\text{Accuracy} = \frac{\text{No of correctly predicted Records}}{\text{Total No of Records}}$$

Predicted Top-k	Accuracy
Top-3	100%
Top-4	95.38%
Top-5	76.15%

Fig10: Accuracy table

V. CONCLUSION

The restaurant recommender system based on cuisine search is developed based on kNN based MapReduce approach. This can be useful to individual users. The methodology proposed may be useful to any person who visits an unknown place and can be helpful to identify the best restaurants of his or her choice. The use of MapReduce in our paper is a novel idea because huge records can be processed with few milli-seconds. The choice of k-NN makes use of the machine learning concept to identify the nearest Neighbors. In the future, the paper can be extended to group users.

REFERENCES

1. Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and Philip S. Yu, "A Survey of Heterogeneous Information Network Analysis", IEEE Transactions on Knowledge and Data Engineering , Volume: 29, Issue: 1, Jan. 2017.
2. C.C. Aggarwal, *Recommender Systems: The Textbook*, Springer International Publishing Switzerland 2016.
3. Zhi-Dan Zhao, Ming-Sheng Shang, "User-based Collaborative-Filtering Recommendation Algorithms on Hadoop" Third International Conference on Knowledge Discovery and Data Mining 2010.
4. Anjali Gautam and Punam Bedi, "Developing content-based recommender system using Hadoop Map Reduce", Journal of Intelligent & Fuzzy Systems 32 (2017) 2997–3008, 2017.
5. Bidyut Kr. Patra, Raimo Launonen, Ville Ollikainen, Sukumar Nandi, "A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data", KNOSYS 3090, 12 March 2015.
6. Chenyang Li Kejing He "CBMR: An optimized MapReduce for item-based collaborative filtering recommendation algorithm with empirical analysis", Concurrency and computation Practice and experience Volume 29, Issue10 25 May 2017.
7. Moon-Hee Park, Han-Saem Park, and Sung-Bae Cho "Restaurant Recommendation for Group of People in Mobile Environments Using Probabilistic Multi-criteria Decision Making", APCHI 2008, LNCS 5068, pp. 114–122, 2008.
8. Glen Jeh, Jennifer Widom,"SimRank: A measure of Structural-Context similarity" in KDD, pp. 538–543, 2002.
9. Peixiang Zhao, Jiawei Han, Yizhou Sun," P-Rank: a Comprehensive Structural Similarity Measure over Information Networks". CIKM'09, Hong Kong, China. November 2–6, 2009.
10. Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, Tianyi Wu, "PathSim: Meta path Based TopK Similarity Search in Heterogeneous Information Networks" *Proceedings of the VLDB Endowment,* Vol. 4, No. 11 2011.
11. C. Shi, X. Kong, Y. Huang, S. Y. Philip, and B. Wu, "HeteSim:A general framework for relevance measure in heterogeneous networks" , IEEE Transactions on Knowledge & Data Engineering, vol. 26, no. 10, pp. 2479–2492, 2014.
12. Sadhana Kodali, Madhavi Dabbiru, Kamalakar Meduri,"Constraint based approach for minging Heterogeneous Information Networks", 6th IEEE IACC 2016, 27th -28th February 2016.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication