Implementation of Knowledge based Collaborative Filtering and Machine Learning for E-Commerce Recommendation System

Mahesh Kumar Singh1, Om Prakash Rishi2, Akhilesh Kumar Singh3, Pushpendra Singh4, Pushpa Choudhary5

1,3,5 Department of Information Technology, G L Bajaj Institute of Technology and Management Greater Noida, U P, India.
2Department of Computer Science and Informatics. University of Kota, Kota, Rajasthan, India.
4Department of Information Technology, Indraprastha Engineering College, Ghaziabad, UP, India

e-mail: 1maheshkrsg@gmail.com, 2omprakashrishi@yahoo.com, 3akhileshaks@gmail.com, 4pushpendra.singh1@gmail.com, 5pushpak2728@gmail.com

Abstract. This is the era of I-way. The development of high-speed computing and huge storage devices change the working culture of human. It affects the traditional business processes and shifted towards online business. It creates huge problems like overload and irrelevant information which are the causes of confusion both customers as well as enterprise. Recommendation system solves these problems. Design and development of efficient system is one of the key areas of the recent researchers. Collaborative filtering (CF) and content-based filtering algorithms are widely used in the implementation of such system. Collaborative used user’s features while content-based used item’s features. Most of the CFs are rating or review based processed homogeneous information. In this paper we proposed knowledge-based collaborative filtering algorithm for large data set that uses various activities done by users during interaction of item through E-commerce web site like clicks, select and purchase. The performance of the system is compared with the base models using real time Amazon E-commerce dataset using precession, recall and NDCG evaluation parameters in various combinations of activities performed by users on items.

Keywords: Web Recommendation System (WRS), Web Artificial Intelligence, Web Information Technology. Collaborative Filtering (CF), Knowledge base, Knowledge graph, Web Usage Mining. Web Engineering Applications.
1. INTRODUCTION

The Internet has transformed the style of traditional ways of business, almost every company wants to create its own website for helping and doing their business. Since Internet provides a very large marketplace hence every customer is faced with multiple choices. Suppose a customer looking to read a book without any specific area, there are many books of same variety, therefore customer spent a lot of time for searching relevant book. If there is any site or app that provides relevant book to the customer as he/she read previously then it saved so many times of the customer. This feature of that web site is known as recommendation system.

Anciently, a person bought product only suggested his/her friends or relatives. This is the traditional way of purchasing when there was any doubt about the product, but this is the era of I-way that circle has expanded to include online sites that utilize some sort of recommendation engine [1]. A recommendation engine figure 1 uses various algorithms to filter and recommends the most relevant products to the customers on the basis of his/her past behaviors that is it recommends products which the user might be likely to buy.

Some popular websites that are using recommendation are This is the era of I-way. The development of high-speed computing and huge storage devices change the working culture of human. It affects the traditional business processes and shifted towards online business. It creates huge problems like overload and irrelevant information which are the causes of confusion both customers as well as enterprise. Recommendation system solves these problems. Design and development of efficient system is one of the key areas of the recent researchers shown in table 1.

![Figure 1 Architecture of Recommendation Engine](image)

Almost every collaborative filtering uses unstructured data such as ratings, reviews are images to profile the users for personalized recommendation. In this paper we can extends the power of collaborative filtering (CF) using large scale structured heterogeneous user behavior data. The main building block of proposed CF combined the traditional CF with knowledge base. The behavior of the users can be represented by directed graph called knowledge graph.
Table 1 Some popular sites that used recommendation system

1.1 Knowledge graph

The representation of relation between customer and product is denoted by a directed graph called knowledge graph [2,3]. It is a directed graph of triplets (subject, predicate, object) called SPO. Subject, object, and predicate, subject and object are entities and predicate show the relationship between these entities [4]. Entities are denoted by nodes and relationship by edges.

“A customer C1 buy a product P1 of category cat1 of brand b1 fall in price range r1 and Customer c1 select product p2 of brand b2 categories cat1 of price range r2” SPO of this statement is listed in Table 2.

Table 2 Subject Predicate Object list of the given statement

| Sr. no | Site     | User | Item          | Description                                                      |
|--------|----------|------|---------------|------------------------------------------------------------------|
| 1      | LinkedIn | Member | Members or Jobs | Members are interested in another members or jobs                 |
| 2      | Facebook | Member | Members       | Members are interested in another members                         |
| 3      | Amazon   | Member | Products or books | Members are interested in products                                 |
| 4      | Netflix  | Member | Movies or Story series | Members are interested in watching movies or story series.       |
| 5      | Flipkart | Member | Products      | Members are interested in products                                |

Combining all SPO triplets together to form a multi directed graph of a given statement as shown in figure 2.
The other activities of the customer can be written as “A customer C1 also view a product p3 of brand b3 of category cat1 fall in price range r1. Customer C1 also buy product p4 of brand b2 of category cat1 fall in price range r2.”

1.1.1 Construction of Knowledge Base. During the construction of knowledge base [5], it is mandatory to consider some parameters like completeness, accuracy, and quality of data which determine the usefulness of knowledge base. There are four major groups of knowledge base construction methods, curated method, collaborative method, automated semi-structured method and automated unstructured method.

1.1.1.1. Curated method. In this method triplets are created manually by a closed group of experts. The accuracy of curated knowledge base is very but this method is not scalable due dependency on human experts.

1.1.1.2. Collaborative method. In this method triplets are created manually by the open group of volunteers. This method is widely used in Wikipedia and Freebase and better scalable, but it also has some limitations due to this growth of Wikipedia has been slowing down.

1.1.1.3. Automated semi-structure method. In this method triplets are extracted automatically from semi-structured text by making some rules. This method is used in Wikipedia infoboxes which has large and highly accurate knowledge graph like YAGO [6] and DBpedia [7], but semi-structure text cover very fraction of information stored on the web.

1.1.1.4. Automated unstructured Method: In this method triplets are created automatically from unstructured text using machine learning and natural languages processing. This method tries to read the web extracts facts from the natural language text of web pages like Nell and Knowledge Vault.

Knowledge graph is similar as knowledge base which is classified into schema based and schema free. Some popular schema-based knowledge base is listed in table 3.

Schema based approach uses predefined and globally unique identified entities and relations in a fixed vocabulary while in schema free approach entities and relations are identified using open information extraction techniques.

| Name of Graph | Entities | Relations | Facts  |
|---------------|----------|-----------|--------|
| Freebase [8]  | 45 M     | 360000    | 735 M  |
| Wikidata[9]   | 19 M     | 1735      | 68 M   |
| DBpedia[7]    | 5.2 M    | 1467      | 638 M  |
1.1.2. Learning of knowledge graph. Let \( E = \{e_1, e_2, e_3, \ldots, e_n\} \) be the set of all entities (that is all the subject and object entities) and \( R = \{r_1, r_2, r_3, \ldots, r_m\} \) be the set of all relations in the knowledge graph. For each possible triplet \( X_{i,k,j} = (e_i, r_k, e_j) \) over this set of entities and relations as a binary random variable \( Y_{i,k,j} \in \{0,1\} \), that is if the relation exists then it 1 otherwise 0. All possible triplets in \( E \times R \times E \) can be grouped in a third order tensor \( Y \in \{0,1\}^{N_e \times N_r \times N_e} \) which contains

\[
X_{i,k,j} = \begin{cases} 
1 & \text{if triplet} \ (e_i, r_k, e_j) \ \text{exists} \\
0 & \text{otherwise} 
\end{cases} \ldots \ldots (1)
\]

Let \( e_h, e_t \) are the head and tail entity in the knowledge graph \( r_k \) is the edge between them then \( e_t \) can be related by \( e_h \) as \( e_t = \text{transpose}(e_h, r_k) = e_h + r_k \) by applying this relation for all nodes relation among the nodes are easily calculated.

2. RELATED WORK

Recommendation System try to identify the user’s interest in the specific domain of contents based on their previous experiences. When a user interacts with the E-commercial site he/she offers a set of implicit or explicit information like clicks, rating, comments etc. about his/her tastes. Recommendation systems are categorized into two main categories personalized [10,11] and non-personalized. Personalized system uses history of the customers’ navigations/behaviors like content based [12], collaborative filtering [13], PageRank [14] in social network analysis etc., while non-personalized system does not require any historical data it used the characteristics of the products like popularity based.

2.1. Content based filtering system

This system is designed to recommend the products on the basis of user’s past preferred order. It saves all the information related to each user in a vector form known as profile vector and another vector, product vector which contains all the information related to the products. This algorithm finds the cosine of the angle between the profile vector and product vector. It uses traditional classification and clustering techniques such as Support Vector Machine [13] or Nearest Neighbors algorithms [12]. There are two types of user implicit and explicit. Those updated their information automatically by the system are called implicit users while some give their feedback to the system in the given range are called explicit users. According to Aggarwal [1] it has some drawbacks like accuracy of the system is highly dependent on the specific application that is used for features of items, over specialization and training size. There are three major limitations of content-based filtering recommendation technique overspecialization, cold start and limitations of contents.
2.2. Collaborative filtering algorithm

It uses user behaviors for recommending items. This is the most commonly used algorithms in the industry since it is not dependent any additional information. There are two types of collaborative filtering techniques, memory based collaborative filtering and model based collaborative filtering.

Memory based collaborative filtering [15] used item based and user-based approaches, recommendations are generated on the preferences of nearest neighbors [ 16], while model based collaborative filtering used matrix factorization approach such as SVD, Tensor factorization [17] it is widely used in order to predict most preferable product wish to purchase by the customer. Graph based or social network-based recommendation system [18] utilized information available from social network like user preferences, influences from friends of social network to overcome the problems of cold start and data sparsity problems or recommendation systems.

3. DESIGN ARCHITECTURE OF PROPOSED MODEL

The architecture of proposed model (see fig 4) contains 4 basic components user that used the system by giving some data to the system and system provided a listed of item in his recommendation list. Event data pre-processing transformed the data as per the system requirements. Ranking algorithm that generated the rank score of each item based on user’s preference, and Matching algorithm that measured the similarity among items and users using item-item and user-user similarity methods respectively.

![Figure 4 Architecture of Proposed Model](image)

3.1. Input Data

Input data sets are collected from Amazone[16] E-commerce web site. It contains various activities performed by users during navigation of the items. Let m be the number users and n be the number of items then there are m x n possible activities are possible, but there are three activities (say view, add_to_cart and transaction) of information used in system as mentioned. The user and item activities is represented by m x n preference matrix (see eq.(2))
3.2. Data pre-processing

Input datasets contain so many attributes in different domains. Every input data set consists of individual data objects, all the data sets have common properties like type of the data object, size, dimensionality sparsity and abstraction. There are some methods which are used to process the data before use of proposed recommendation system.

3.2.1. Feature selection. There are many attributes in the input from users and items but, method of dimension reduction of the datasets which is primarily used to remove redundant and irrelevant attributes from the datasets. The system consider c_id, ip_address and session attributes from the customer dataset, P_id, P_cat from product dataset and P_view, P_select, and P_buy from relation datasets and all others are removed.

3.2.2 Data binarization. It is the method to provide the values of the attributes here the values of the relations are considered for computation of preference score of the product hence the values of the attributes like P_view, P_select and P_Buy are binarized that either 0 for no and 1 for yes, therefore the preference matrix will be.

\[ C_{ij} = \begin{bmatrix}
  c_{11} & c_{12} & \ldots & c_{1n} \\
  c_{21} & c_{22} & \ldots & c_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  c_{m1} & c_{m2} & \ldots & c_{mn}
\end{bmatrix} = m \times n \text{ matrix } \quad (2) \]

3.3. Computation of preference or ranking score using knowledge recommendation system (KRS)

The preference or ranking score of the product is computed on the number of customers participated in the events in a specific period of times.

3.3.1. Event Database. It is the collection of customer events performed on different categories (P_cat) of products (see Table-4). Suppose \( C^i_j \) shows the customer i that buy the product j (i.e. a product j which is buy by customers i), \( C^i_j \) shows the customer i that select the product j (i.e. product j is only selected by the customer i) Similarly \( C^i_j \) shows the customer i that view the product j (i.e. the product j is view by the customer i). The preference order between the products is measure as product buy>product _select>product_view. The customer’s preference of the product can be represented in matrix form as \( C^i_j \) it means customer i prefer in terms of buy or select or view product j. If there are m number of customers and n number of products then \( i=1,2,3,4,\ldots\ldots.m \) and \( j=1,2,3,4,\ldots\ldots.n \). The product preference matrix \( (C^i_j) \) can be written by \( m \times n \) matrix (2).

| Name of Attribute | Data Type | Description                        |
|-------------------|-----------|------------------------------------|
| P_view            | Number    | Product view of click              |
| P_select          | Number    | Product selected or add_to_cart    |
| P_buy             | Number    | Product bought or transaction      |
3.3.2. Customer data. The user database (see Table 6) contains specific data like c_id, url, location of the customer, navigational details, timestamp etc. The web server logs are represented as the tuple \{u_id, ip_address, url, timestamp, location\}, that represent the details of web access user, pages requested to access, corresponding browser time and the location of the user. But c_id of the customer is beneficial for recommendation. The u_id with the url which are visited by the customer can be set up by the association matrix.

### Table 5 Product Data base

| Name of Attribute | Data Type | Description |
|-------------------|-----------|-------------|
| P_ID              | Varchar   | Unique Identification number of the Product |
| P_cat             | Text      | Category of product |
| P_brand           | Text      | Name of the brand of a product. |
| P_name            | Text      | Name of the product |
| Price             | Real      | Price of product |
| Unique_page       | Number    | Number of unique pages visited by the user per product. |
| P_session         | Time      | Time spent by user for a particular product ID |

\[
M_{rs} = \begin{bmatrix}
  v_{11} & v_{12} & \cdots & v_{1s} \\
  v_{21} & v_{22} & \cdots & v_{2s} \\
  \vdots & \vdots & \ddots & \vdots \\
  v_{r1} & v_{r2} & \cdots & v_{rs}
\end{bmatrix}
\]  

Here \(v_{ij}\) is the browser visiting information is a particular time that user \(i\) visited the first \(j\) pages (product).

The URL gives the idea about the product like product_id and product_category (see Table-5). The user’s information with the product’s information is used for the recommendation.

| Name of Attribute | Data Type | Description |
|-------------------|-----------|-------------|
| C_ID              | Number    | Unique identification number of any user |
| Age               | Number    | Age group of the customer |
| Gender            | Text      | Gender group |
3.3.3 Method. The dataset consists of product details, customer details, page detail, action, and categories, brands of products. The relation between user and item is denoted by the triples (Ci, Rk, Pj), this shows that customer i is related with particular relation k with product j. The first aim is to find the order pair (Ci, Pj) bought, select, and view list of customers then we find the order pair (Ck, Pj) that is list of customers related with that product Pj.

The customer’s preference rank can be calculated by the formula

\[ R_{ij} = X \frac{C_{ij}^b}{\sqrt{\sum_{j=1}^{n} (C_{ij}^b)^2}} + Y \frac{C_{ij}^s}{\sqrt{\sum_{j=1}^{n} (C_{ij}^s)^2}} + Z \frac{C_{ij}^g}{\sqrt{\sum_{j=1}^{n} (C_{ij}^g)^2}} \]  \hspace{1cm} (4)

Where X, Y and Z are the weight adjusting coefficients corresponding to three different relations of shopping all are not same as the preference of these steps the values of Z>Y>X. At value X=0.25, Y=0.5 and Z=1 is suitable for better results. Hence this formula finds the preference score of the products higher value means higher the score.

3.4. Matching algorithm for recommendation (Association Rule Mining)

The rule mining is used to create the knowledgeable bunches data of similar category that it associates the customer profiles with each product. That is valuable for the recommendation both customer as well as for the enterprise. It is clear that there are three categories of the products hence association rules can be generated from three different kinds of transaction sets, called buy transaction set, selected_but_not_buy and also_view transaction set. For each transaction from the web logs there are three steps for the association rules. First deals the minimum support and minimum confidence, second replace each product in the transaction set with its corresponding categories of products and in third step generate association rule for each transactional set using Apriori algorithm. It can be given by a matrix \( P=P_{kl} \) called product matrix, where \( k=1,2,...,n \) is the total number of products and \( l=1,2,3 \) or 4 define in the product matrix \( P_{ij} \) that represents the association degree among the product categories in different transactional steps.

\[
P_{kl} = \begin{cases} 
1.0 & \text{If product is bought} \\
0.50 & \text{If product is selected} \\
0.25 & \text{If product is view} \\
0.00 & \text{otherwise} 
\end{cases} 
\] \hspace{1cm} (5)

The relation among these three parts of information is denoted by the triples (Ci, Rk, Pj), this shows that customer i is related with particular relation k with product j. The first aim is to find the order pair (Ci, Pj) bought, select, and view list of customers then we find the order pair (Ck, Pj) that is list of customers related with that product Pj.
There are two different matrices customer preference matrix and product association matrix before recommending it is to find the matching scores for each product associated with each customer this score gives the idea closeness of product with the customer.

### 3.4.1. Customer-Customer collaborative filtering.

This algorithm finds the similarity score between customers. Based on this similarity score, it then picks out the most similar users and recommends products which these similar customers have liked, selected or bought previously.

A user based Collaborative Filtering technique works based on a set of customers \( u \) have the same relation to the product by aggregating these relations using the formula (6).

\[
    r_{u,j} = \frac{1}{K} \sum_{k \in N_u} \text{Sim}(u,k) \cdot r_{k,j} \quad \ldots \ldots \quad (6)
\]

Where \( N_u \) is the set customers of which \( K \) customers have the similar interest target to customer \( u \). \( \text{Sim}(u,k) \) shows similarity between customer \( u \) and \( k \) predefined customers, and \( r_{k,j} \) represents the rating given by \( k \) users to the product \( j \).

### 3.4.2. Product-Product collaborative filtering.

In product based collaborative filtering considers the similarity among the products or services. It is supposed that similar products are related in a similar way by the same customer. Hence the products recommended to the customer \( u \) are scored or ranked by aggregating the similarity of the different customers and the customer \( u \) related in the past. It is possible to compute similarity score by the given formula (7).

\[
    r_{u,j} = \frac{1}{K} \sum_{k \in N_j} \text{Sim}(j,k) \cdot r_{u,k} \quad \ldots \ldots \quad (7)
\]

Where \( N_i \) denotes the set of products or items neighbor to \( j \). \( \text{Sim}(j,k) \) is similarity value.

### 4. EXPERIMENTS AND RESULT ANALYSIS

Experiments are performed on Amazon E-commerce datasets [19] there are five kinds of sub-datasets automotive, mobile phones, home appliances, movies, books. The behavior of data i.e. relations (events) like view, select and buy represent interaction that were collected over period of 4.5 months. In the original data there are 27,561 events including 26,643 only view, 69,332 only select and 22,457 buys produced by 140758 unique customers and 8885 unique products.

The statistics of these datasets are summarized in the table. We consider Top N recommendation measurement including Precision, Recall, Hit Ratio, NDCG [20] for evaluating the model and baselines. Fist three methods are used to evaluate the quality of recommendation system and last method is used for accuracy and ranking positions of the correct products in output list. There are so many relations among the entities of the knowledge graph, but this paper considers only three relations.

- **Buy relation:** This relation shows that a customer \( c \) bought the product \( p \).
- **Select relation:** This relation shows that the customer \( c \) add_to_cart the product \( p \).
- **View relation:** This relation shows that a customer \( c \) visits the product \( p \) page.

#### 4.1. Computation of baselines

We use following methods for baselines performance comparison.

- **Bayesian Personalized Ranking (BPR)** [21]. It is a popular method in Top N recommendation used matrix factorization as the prediction component. It is based on the triplet \((u,i,j)\) where user \( u \) interact with item \( I \) but not item \( j \). Relationship between item \( I \) and \( j \) with respect to user \( u \) can be given by the formula

\[
    x_{ui} = x_{uj} > 0 \quad \ldots \ldots \quad (8)
\]
4.1.2. Hidden Factors and Topics (BPR_HFT) [22]. It is a method used for textual reviews; we use HFT under BPR pair wise ranking framework for fair comparison.

4.1.3. Visual Bayesian Personalized Ranking (VBPR) [23]. This method is used for recommendation with images.

4.1.4. Deep Convolutional Neural Network (DCNN) [24]. It is a review based deep recommendation method to jointly model the users and the products.

4.1.5. Joint Representation Learning (JRL) [19]. It is a model which can leverage multi-model information for Top-N recommendation.

4.2 Ranking accuracy

It deals the levels of utility of the recommended product or service with respect to the ranking proposed by the user. Discounted Cumulative Gain (DCG) is very popular matrix for evaluating the ranking accuracy. The Normalized DCG [27] is defining as follows.

\[ DCG = \frac{1}{m} \sum_{u=1}^{m} \sum_{j \in I_u} \frac{g_{uj}}{\log_2(v_j + 1)} \quad \text{NDCG} = \frac{DCG}{IDCG} \quad (9) \]

Where \( m \) denotes the total number of users in the test dataset \( D_e \), \( I_u \) is the set of products/services liked by user \( u \), \( v_j \) is position of \( j \) in the recommended list, \( g_{uj} \) represents the utility gain given by the user \( u \) to the product \( j \), and IDCG is the ideal value computed on the basis of real value using same formula as DCG. Another way to evaluate the accuracy of the relevant list is to consider the tradeoff between the length of the list \( R_L \) and the number of actual relevant products/services for the user. The relevant list \( R_L \) contains true positives \( (t_p) \) but not false negative \( (f_n) \) and false positive \( (fp) \). Hence \( R_L \) and number of relevant products can be computed in terms of Precision and Recall [10] as follows.

\[ \text{Precision} = \frac{t_p}{t_p + fp}, \text{Recall} = \frac{t_p}{t_p + f_n} \quad (10) \]

In the single matrix it can be summarized in F-measure [10] which can be computed by the following formula.

\[ F-measure = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11) \]

4.3 Settings of parameters

All the parameters used in this filter are initialized in the range \((0,1)\) and updated as per Stochastic Gradient Descent (SGD). The learning rate is determined in the range of \( \{1.0,0.1,0.01,0.001,0.0001\} \) and the dimension is started in the range \( \{10,50,100,200,300,400,500,600\} \) which gives the final learning rate as 0.01 and dimension as 200. For computing the baselines 70% products of each user are consider for training while other are used for testing. The system generated top 10 recommendation for each user from the test dataset.

4.4 Performance Comparison

The performance of proposed filter shown in table 7 and table 8. Table-7 shows the performance comparisons with various base models and table-8 shows the performance with possible combinations of relations. From the experimental result (See fig 5) it is clear that both reviews based and rating based models enhanced the performance of recommendation system but heterogeneous information source-
based model like JRL performs more better than baseline system, which gives the idea that the knowledge based collaborative filtering (KCF) performs more better than that of JRL consistently over five datasets and all evaluation measures which verifies the proposed system.

Table 7 Performance table on top 10 recommendation between baselines and proposed system.

| Data Set | Automotive | Mobile phones | Home appliances | Movies | Books |
|----------|------------|---------------|-----------------|--------|-------|
| **Measurement** | **Recall** | **Prec** | **NDCG** | **Recall** | **Prec** | **NDCG** | **Recall** | **Prec** | **NDCG** | **Recall** | **Prec** | **NDCG** | **Recall** | **Prec** | **NDCG** | **Recall** | **Prec** | **NDCG** |
| BPR      | 2.08      | 1.93         | 2.41           | 3.28    | 0.89      | 3.198        | 1.11      | 0.19      | 0.61      | 4.241    | 1.125      | 2.723        | 2.124    | 1.244      | 2.124         |
| BPR_MFET| 3.221     | 3.221        | 2.118          | 3.221   | 0.703     | 2.233        | 1.19      | 0.195     | 0.61      | 4.241    | 1.125      | 2.241        | 2.145    | 1.252      | 2.145         |
| NWHPR   | 0.183     | 0.392        | 0.702          | 0.188   | 0.188     | 0.987        | 0.17      | 0.17      | 0.61      | 2.27     | 0.392      | 1.912        | 0.596    | 0.785      | 1.825         |
| D2SN    | 5.092     | 3.092        | 3.601          | 5.092   | 2.254     | 3.231        | 5.201    | 1.201     | 1.201     | 5.201    | 1.201      | 1.201        | 2.501    | 1.405      | 1.254         |
| JRL     | 7.439     | 2.089        | 5.473          | 7.439   | 4.45      | 5.112        | 3.14      | 1.45      | 1.11      | 10.45    | 5.145      | 8.045        | 3.241    | 1.225      | 4.455         |
| KCF     | 8.124     | 1.21         | 5.708          | 8.124   | 5.935     | 5.565        | 8.124    | 1.21      | 1.11      | 10.45    | 5.565      | 8.124        | 5.935   | 1.21       | 5.708         |

Table 8 Performance on top 10 recommendation when incorporating between varieties of relational structures used in knowledge graph. The final result is significantly than all other models.

| Relations  | Automotive | Mobile phones | Home appliances | Movies | Books |
|------------|------------|---------------|-----------------|--------|-------|
| Measurement | **Recall** | **Prec** | **NDCG** | **Recall** | **Prec** | **NDCG** | **Recall** | **Prec** | **NDCG** | **Recall** | **Prec** | **NDCG** | **Recall** | **Prec** | **NDCG** | **Recall** | **Prec** | **NDCG** |
| Buy       | 5.395      | 2.019        | 5.51            | 5.395   | 0.995     | 5.395        | 2.395    | 1.39      | 2.273     | 4.254    | 1.224      | 2.242        | 3.241    | 1.245      | 2.242         |
| Select    | 3.576      | 1.784        | 2.418           | 3.576   | 0.784     | 3.576        | 3.576   | 1.784     | 2.418     | 4.45     | 1.245      | 2.454        | 3.241    | 1.245      | 2.454         |
| view      | 1.594      | 1.303        | 1.701           | 1.594   | 0.594     | 1.594        | 1.594   | 1.303     | 1.701     | 1.594    | 0.594      | 1.701        | 1.594   | 0.594      | 1.701         |
| Buy+Select| 5.692      | 2.187        | 3.803           | 5.692   | 2.187     | 3.803        | 5.692   | 2.187     | 3.803     | 5.692    | 2.187      | 3.803        | 5.692   | 2.187      | 3.803         |
| Buy+View  | 4.655      | 2.095        | 3.657           | 4.655   | 2.095     | 3.657        | 4.655   | 2.095     | 3.657     | 4.655    | 2.095      | 3.657        | 4.655   | 2.095      | 3.657         |
| view+Select| 3.631     | 1.99        | 2.708           | 3.631   | 1.99      | 2.708        | 3.631   | 1.99      | 2.708     | 3.631    | 1.99       | 2.708        | 3.631   | 1.99       | 2.708         |
| All(KCF)  | 6.129      | 2.295        | 4.704           | 6.129   | 2.295     | 4.704        | 6.129   | 2.295     | 4.704     | 6.129    | 2.295      | 4.704        | 6.129   | 2.295      | 4.704         |

Figure 5 Performance of top 10 recommendation when incorporating between varieties of relational structures used in Knowledge Graph
5. CONCLUSION

This paper discussed the concept of knowledge graph its learning and creation. Create a knowledge-based collaborative filtering that processed the heterogeneous information which are unstructured by using the concept of knowledge graph that is a directed graph it converted into structured form. The triplet tuple relation between user and the product played a vital role in the development of proposed CF. Experimental results used real world datasets for performance measurements of various filters used in recommendation system based on rating, review and heterogeneous information. From the result it is clear that the performance of proposed filter is much better than the discussed filters, therefore we conclude that the proposed CF is better for creation of recommendation system.

REFERENCES

[1] Aggarwal C.C. 2016 “Recommender Systems”, Springer.
[2] Catherine R., K. Mazaitis, M Eskenazi, W Cohen: “Explainable Entity Based Recommendation with Knowledge Graph”, RecSys 2017.
[3] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: a collaboratively created graph database for structuring human knowledge,” in Proceedings of the 2008 ACM SIGMOD international conference on Management of data. ACM, 2008, pp. 1247–1250.
[4] L. De Raedt, Logical and relational learning. Springer, 2008.
[5] Mahesh Kumar Singh, Om Prakash Rishi, Sumit Wadhwa “Application of Page Ranking Algorithm Based on Numbers of Link Visits in Web Recommendation System for Online Business” 4th International Conference on Green Computing and Engineering Technology (ICGSET-2018), on 17th -19th August 2018, at Aalborg University, Nebels Hohrs Vej 8, Esbjerg, Denmark.
[6] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr, and T. M. Mitchell, “Toward an Architecture for Never-Ending Language Learning,” in Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010). AAAI Press, 2010, pp. 1306–1313
[7] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “DBpedia: A Nucleus for a Web of Open Data,” in The Semantic Web. Springer Berlin Heidelberg, 2007, vol. 4825, pp. 722–735.
[8] Ruining He, Wang-Cheng Kang and Julian McAuley “Translation based Recommendation: In RecSys. ACM 2017.
[9] F. M. Suchanek, G. Kasneci, and G. Weikum, “Yago: A Core of Semantic Knowledge,” in Proceedings of the 16th International Conference on World Wide Web. New York, NY, USA: ACM, 2007, pp. 697–706.
[10] Mahesh Kumar Singh, Om Prakash Rishi, Sumit Wadhwa” Knowledge Generation through Web Mining Techniques for E-business Recommendation System” 1st International Conference on Communication and Computing (ICCCT-2018), on 7th and 8th Sept 2018, at GLBITM Gr Noida, UP India.
[11] M. K. Singh, O. P. Rishi, S. Awasthi, A. P. Srivastava and S. Wadhwa, "Classification and Comparison of Web Recommendation Systems used in Online Business." 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM), Dubai, United Arab Emirates, 2020, pp. 471-480.
[12] P. Lops, M. De Gemmis, G. Semeraro:” Content-based recommender systems: state of the art and trends”, in: Recommender Systems Handbook, Springer, 2011, pp. 73–105
[13] Z. Xia, Y. Dong, G. Xing “Support vector machines for collaborative filtering”, in: Proceedings of the Forty-fourth Annual Southeast Regional Conference, ACM, 2006, pp. 169–174.
[14] L. Page, S. Brin, R. Motwani, and T. Winograd. “The PageRank citation ranking: Bringing order to the Web”. Technical report, Stanford Digital Libraries SIDL-WP-1999-0120, 1999.
[15] Al-shamri M.Y.H. “Expert System with Applications Power Coefficient as a similarity measure for Memory based collaborative recommender systems “Expert System with Application 41(13), pp 5680-5688, 2014.
[16] M.Y.H. Al-shamri,” Expert System with Applications Power Coefficient as a similarity measure for Memory based collaborative recommender systems” Expert System with Application 41(13), pp 5680-5688, 2014

[17] K. Miyahara, M. Pazzani :” Collaborative filtering with the simple Bayesian clas- sifier”, Proceedings of the Eighteenth Pacific Rim International Conference on PRICAI 2000 Topics in Artificial Intelligence, 2000, pp. 679–689.

[18] Mahesh Kumar Singh, Om Prakash Rishi “Event Driven Recommendation for E-commerce using Knowledge based Collaborative Filtering Technique. ” Scalable Computing: Practice and Experience, ISSN 1895-1767. Volume 21, Issues 3,(2020-SCPE), pp. 369–378, DOI 10.12694/scpe.v21i3.1709

[19] J. McAuley, Leskovec, J. “Hidden:factors and hidden topics: understanding rating dimensions with review text”. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 165–172.

[20] H. Seifoddini and M. Djassemi: “The production data-based similarity coefficient versus jaccard’s similarity coefficient,” Computers & industrial engineering, vol. 21, no. 1, 1991, pp. 263–266.

[21] S Rendle. C Freudenthaler, Z. Gantner, L. Schmidt-Thieme. “BPR: Bayesian personalized ranking from implicit feedback”. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–21 June 2009; pp. 452–461.

[22] Y. Zhang, Q. Ai, X. Chen, W.B. Croft:” Joint representation learning for top-n recommendation with heterogeneous information sources”. In Proceedings of the 2017 ACM Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1449–1458.

[23] He, R., & McAuley, J. (2016). VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. Proceedings of the AAAI Conference on Artificial Intelligence, 30(1). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/9973

[24] Mahesh Kumar Singh, Om Prakash Rishi, Anukrati Sharma, Zaved Akthatar “Knowledge Extraction Through Page Rank Using Web Mining Techniques for E-business: A Review “ a chapter in” Maximizing Business Performance and Efficiency through Intelligent Systems”