ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2018.0229

CROSS DOMAIN RECOMMENDATION USING VECTOR MODELING AND GENRE CORRELATIONS

Mala Saraswat, Shampa Chakraverty, Sakshi Garg, Sweta Nandal and Vibhav Agarwal
Division of Computer Engineering, Netaji Subhas Institute of Technology, India

Abstract

Recommender systems are basically information retrieval systems that offer guidance to users in making individual decisions related to choosing items based on personal interests. On Internet, there are infinite numbers of results for a particular query like movies, music, books, clothes etc. Sorting through every result is very tedious and time-consuming. Recommender system is very important application of data science and machine learning. They make the job of recommendation and prediction of preferences of users very simple. There are many limitations in classical recommender system because they provide recommendations in single domain only. With proliferating e-commerce sites and limitations in collaborative and content based recommender systems, cross domain recommender system are now widely in use. They can address the data sparsity and cold start problem by utilizing data from other related domains. In this paper, we propose recommendations across different domains by combining the benefit of plot keywords extracted from storyline and genre details from the two entertainment domains. We illustrate the working of our proposed CDR scheme using the movie as source domain and book as target domain.

Keywords:

Cross Domain Recommender System, Cold Start Problem, Keywords, Vector Modeling, Genre Correlation

1. INTRODUCTION

Recommender systems play out the capacity of data separating and provide customized personal recommendations to users. In the computerized time, there are boundless numbers of results for a specific search like films, books and so on. In this situation, the user needs to deal with numerous results to get the desired outcome which is very laborious. A need to deal with this issue and enhance client experience gave rise to recommender systems. Since the interests and tastes of peoples don't change much amid a short interval of time, so by appropriate analysis of user history one can foresee their interest and give customized suggestions. To achieve this, numerous sites give arrangements for feedbacks, remarks, and reviews to know users interests [1]. Recommender systems are utilized in different fields like motion pictures, music, clothes, books, financial services, social websites like facebook, twitter, Instagram etc., online dating, food, gadgets, online shopping like Amazon, Flipkart etc., travel [2] and so forth. Their significance can be acknowledged from the fact that 30% of Amazon income is generated by them and just about 75% of clients watch Netflix suggested motion pictures. Recommendation systems use tags, reviews, ratings, feedbacks, genres etc. to provide recommendations.

Recommender systems frameworks are broadly classified into two unique classes i.e. collaborative filtering and content filtering. Collaborative filtering makes utilization of the fact that users who had basic interests and inclinations in past will have same interests now. It discovers the closest neighbors of the user and with the assistance of their rating-pattern similarities and information provide recommendations [3]. One of the basic limitation of Collaborative filtering based recommender system is the issue of cold start i.e. any outcome or deduction can't be drawn for another user that has not yet rated anything since no similar neighbors be found [4]. Content based recommender system utilizes the fact that the things user liked in the past will most presumably be liked at this point. It includes i) examining client history and reviews ii) preparing user profile iii) suggesting the best match [5]. This approach suffers from new user cold start issue since there isn't sufficient data accessible to assemble client profile. Likewise if adequate content of an item isn't accessible, suggestions can't be made. Cross Domain Recommender systems address these weaknesses by using data from other related domains as there exists a few connections and correlations between them [6]. CDR help in giving customized proposals over various domains like music, books, films by dissecting users' choices crosswise over various fields.

In this paper, we have exploited the benefit of both collaborative and content-based filtering. We have used collaborative filtering to calculate genre correlations to form genre correlation matrix and content-based filtering by utilizing keywords extracted from storyline of items for computing the similarity score between the domains. For the experiments, we have chosen movies as source domain and books as target domain since both movies and books have a plot, a storyline, a topic, characters, dialogues, genre which relates them. Indeed, numerous movies depend on books. We have utilized the evaluations provided by the users, genre information about books and movies and keywords for providing suggestions. Keywords are extracted using information gathered from user reviews and storyline provided by the film specialists and directors. Keywords are the words that can be utilized to look through the item. It is not necessary for keywords to explicitly characterize an item rather they just aid in narrowing down the pursuit to 5-10 things. Each movie and book is related with at least one classification or genre. A user genre preference can be of extreme use since it can be safely assumed that if a user prefers a movie of a specific type then he would likewise, like books with comparable types. Another imperative certainty to be considered is that a few genres are more related than others. For example, *action* and *adventure* or *crime* and *thriller* forms more sense than *action* and *social* or *satire* and *philosophy* [7].

In this paper, we framed a genre correlation matrix from our dataset for computing genre classification score. The final similarity score is computed utilizing both the keyword similarity and genre correlation scores. Target items are sorted based on similarity scores so as to recommend items with the top *N* highest score.

1.1 PRIOR WORK

In this section we discuss the previous work in cross domain recommender system. Tobias et al. [8] used generic based framework and semantic net to relate different domains. Yang et al. [4] in his paper tries to solve sparsity problem in collaborative filtering. In his paper the author uses transfer of user-item rating patterns from a dense auxiliary rating matrix in other domains to a sparse rating matrix in a target domain. Cantador et al. [6] surveys state of the art cross-domain recommender systems. Shapira [1] makes use of data from social networking site for the recommendation. She showed that when data is not available for a new user or is sparse, recommendation results from social data are equally as precise as results obtained from user ratings. Berkovsky [9] used content-dependent partitioning of collaborative movie ratings. In this partitioning of ratings is done according to the genre of the movie.

For the recommendation process Roza et al. [10] used semantic similarity measure which is domain independent. Karamollah et al. [11] found a set of *k* nearest neighbors to the target user. Semantic analysis was used to generate user profile and then semantic similarity among users' profile was used. Shampa et al. [12] generated user emotion profile from online content like reviews for CDR. Rui [13] used semantic technologies to enhance slope one. Slope one is a collaborative filtering model which is based on average rating difference. Hwang et al. [7] proposes a genre correlations method for movie recommendation. The proposed algorithm computes the correlation between genres using ratings given by users and provides a ranked list of recommended movies for a target user based on the calculated genre correlations. The limitation of this approach is that it works for single domain only. We improved this method to work for cross recommendation, whereby genre correlation is established between genres of different related domains like books and movies. We further enhanced the approach by incorporating keyword similarity between the keywords of movies and books apart from genre correlation to give better recommendations

2. PROPOSED APPROACH

This section discusses our keyword and genre correlation based approach for cross domain recommender system. The proposed approach can be summarized with the help four modules which combine to give the results. M_1 is Data preparation module wherein keywords are extracted from the datasets. Modules M_2 Keyword Similarity and M_3 Genre correlation process in parallel wherein M_2 computes the similarity score between keywords of source and target domain and M_3 generates the genre similarity score using genre correlation matrix. Both the keyword and genre similarity score are combined in Module 4 to give the final top N recommendations. The Fig.1 shows the block diagram of the proposed approach.

3. DATA PREPARATION

In this module M_1, we extract keywords from storyline from the given dataset. Keywords hold significance because distributional hypothesis [14] suggests that if words mean have same semantic interpretation then they will have similar distribution and occur in related context. We used real dataset for movies and book domain. The Movies dataset [15] consists of several attributes like ratings, keywords, cast, crew, genres etc. So we extracted keywords and genre. We used book crossings dataset [16] that contain the book id and ratings of the book. It does not contain corresponding keywords and genre. For extracting the keywords, we used the book dataset from Carnegie Millon university dataset [17] that contain the storyline and genres for the books. Since the book dataset contains storyline instead of keywords. The keywords are obtained by annotating the storyline by users. Thus after all data cleaning stages 2700 books are obtained for our proposed approach.

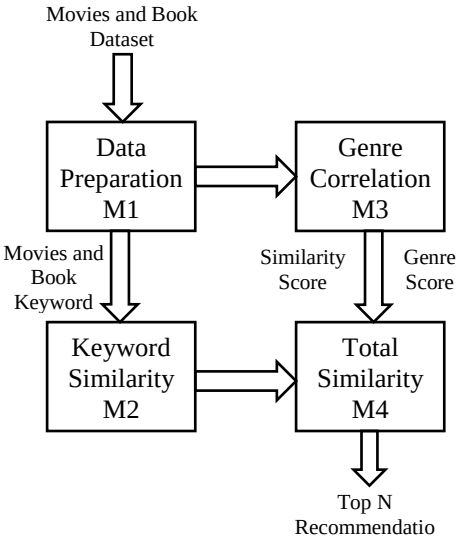

Fig.1. Block diagram for the proposed CDR approach

3.1 KEYWORD SIMILARITY

In this module M_2, we compute book similarity score on the basis of similarity in keywords between books and movies. This module follows three steps as under:

Step 1: In this step, we used vector modeling to convert document (movie and book keywords) into vectors. A corpus is created using all the keywords. Movies and Books are then converted into vectors where number of dimension is equal to total distinct keywords. At the end of this stage every movie and every book is represented as vectors where the dimension is number of distinct word in complete corpus.

Step 2: We then applied Topic Modeling to reduce number of dimensions and extract topics from the documents. LDA transformation model is applied to the keywords of book dataset and movie dataset. This process serves two goals: To make hidden patterns of document visible and converting documents into more semantic way by making use of discovered relationships between words. To convert the documents into more concise form. It helps in improving efficiency and efficacy.

Step 3: In this step similarity between the movies and books is calculated using cosine similarity as shown in Eq.(1). It can be seen that more are the vectors similar to each other, less is the angle between them and more is the value of cos (angle between vectors). We have

considered that cos(θ) cannot be negative between two documents. Calculated cos(θ) is the keyword similarity score. As cos(θ) can only be less than 1, we get normalized score only. Mathematically,

$$Simk = \cos(\theta) = A.B/|A|.|B| \quad (1)$$

3.2 GENRE CORRELATION

The purpose of this module M_3 as shown in Fig.2, is to find genre similarity score on the basis of correlation and ratings. The movie genre as related to book genre is represented as *corr*(*a*,*b*) in the matrix where *a* represent a movie genre and *b* represents a book genre.

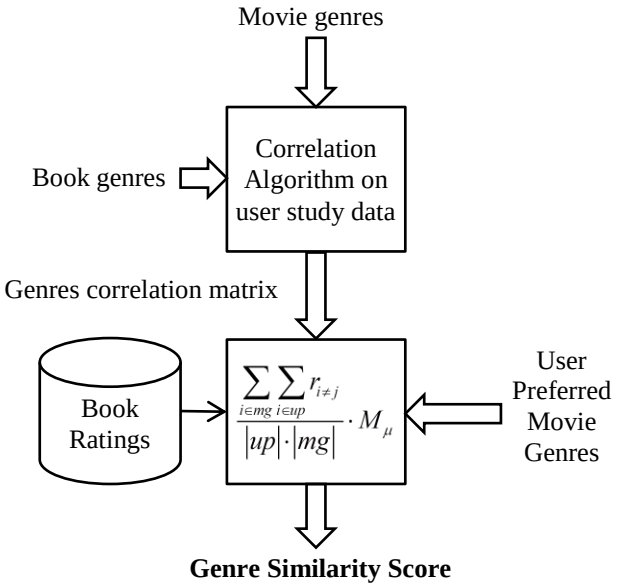

Fig.2. Block diagram representing calculation of similarity score using genres

For computing genre correlation following steps are followed. We have used following steps:

Step 1: Cross domain Genre Correlation Matrix is formed by collecting overlapping dataset wherein users provide his movies and books preferences. We separated our dataset into training and testing data. We used 150 users for testing of model and rests of the users were used for training the model and calculating genre correlation matrix. After training the model we got genre correlations. Let us say, movie *a* has a genre combination (g_1, g_2, g_3) and book *b* has genre combination (h_1, h_2). Then movie genre g1 is chosen as a criterion genre and we change the combinations, starting from h_1 and increasing count by one. We apply similar process for all movies in the dataset. For expressing the correlation, the frequency of g_1h_1 is divided by total frequency of g_1. The Fig.3 shows the pseudocode for generating the correlation matrix.

Step 2: Genre similarity score for each book *R* is calculated by using equation (2) as discussed in [7]. The Eq.(2) utilizes genre correlation and average rating of target domain.

$$R = \begin{cases} \dfrac{\sum\limits_{i\in mg}\sum\limits_{i\in up}\left(r_{i=j}+\dfrac{r_{i\neq j}}{|mg|-1}\right)}{|up|}\cdot M_\mu & \text{if } i\neq j \\ \dfrac{\sum\limits_{i\in mg}\sum\limits_{i\in up} r_{i\neq j}}{|up|\cdot|mg|}\cdot M_\mu & \text{if } i=j \end{cases} \quad (2)$$

where, *up* is set of user preferred movie genres, |*up*| is cardinality of *up*, *mg* is the set of all books with their ratings, |*mg*| is cardinality of set mg, M_μ is the mean rating of book, *r* is genre correlation between i^{th} genre of up and j^{th} genre of mg when genre *i* is not equal to genre *j*.

EvalGenreCorrelation(.)

Input: Movie n genre, Book m genre
Output: correlation matrix *corr*[*i*][*j*]

```
For all k belonging to users
For all i belonging to genre[k].movies
For all j belonging to genre[k].books
corr[i,j]=corr[i,j]+1;
For all i belonging to genre of movies(n)
For all j belonging to genre of books(m)
sum(i)=sum(i)+corr[i,j];
For all i belonging to genre of movies(n)
For all j belonging to genre of books(m)
corr[i,j]=corr[i,j]/sum(i);
```

Fig.3. Pseudocode for Finding Correlation between Genre

3.2.1 Genre Correlation Matrix:

In our approach we used 18 movie genres and 50 books genres. Movie genres used are action, adventure, animation, comedy, crime, drama, documentary, drama, fantasy, historical, horror, mystery, political, romance, science fiction, satire, social and thriller. We used our dataset to create genre correlation matrix. A part of this correlation matrix is shown in Table.1.

Table.1. Snapshot of Genre Correlation Matrix

Genre	Sports	Military	Zombie	Prose	Adventure
Fantasy	20	7	70	70	75
Romance	4	3	4	80	55
Adventure	60	70	59	30	95
Odyssey	35	60	20	75	65
Comedy	10	4	40	35	15
Horror	11	12	88	11	12
Action	23	60	78	11	90

4. TOTAL SIMILARITY SCORE

This module M_4 as shown in Fig.4 combines the keyword similarity score and genre similarity score to find the total similarity score computed using Eq.(3). Based on the similarity score top N items of target domain are recommended to user from source domain.

$$TS = \omega*(GS)+(1-\omega)*Simk \quad (3)$$

where, *TS* is Total Score, *GS* Genre similarity Score and *Simk* is keyword similarity Score and ω is the weight assigned.

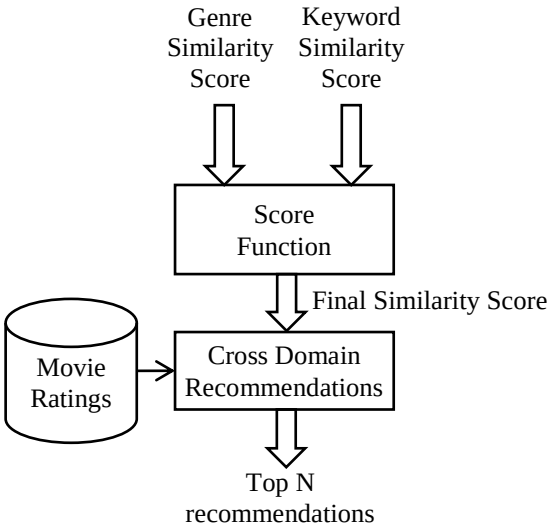

Fig.4. Block diagram representing final recommendation

5. EXPERIMENTS AND RESULTS

For our research, we have used two real-life datasets. For source domain i.e. movies domain we used The Movies dataset from kaggle [15] and for target domain i.e. book domain we have used book CMU dataset [16]. In the Movies dataset, we had around 2 lakh users and 46419 movies. Movies dataset for each movie consists of keywords, genre details, tags, ratings etc. In books dataset, we had around 50 thousand users and approximately 10 thousand books. This dataset contained storyline, genre and author details for every book. Since original books dataset did not contain keywords, we extracted keywords separately by annotations. We selected only those users who rated both movies and books and only those movies and books which were rated by overlapping users for our research. The Table.1, Table.2 and Table.3 shows the characteristics of movie and book dataset respectively.

Table.2. Movies Dataset

Attribute	Meaning
Id	Unique id no. of every movie
Title	Title of movie
Keywords	Keywords representing the movie
Genres	Genres of the movie

Table.3. Books Dataset

Attribute	Meaning
Id	Unique id no. of every book
Title	Title of the book
Storyline	Storyline of the book
Genres	Genres of the book
Author	Author of the book

6. EVALUATING THE PERFORMANCE OF PROPOSED APPROACH

We now summarize the results of our findings. We have used both genre score and keyword score for our final result. We compared precision, recall and F-measure taking different weightage for genre similarity and keyword similarity. We evaluated our approach for various values of ω. We have listed our results varying ω=0.2 as shown in Table.4, ω=0.4, as shown in Table.5, ω=0.6 as shown in Table.6 and ω=0.8 as in Table.7. The Fig.7 compares the precision values for different values of ω. As can be seen from Fig.7 for ω=0.2 we get the best results compared to other weights. The Fig.8 shows precision, recall and F-measure for ω =0.2.

Consider an example where top 10 movies liked by user1 are considered. The Fig.5 shows a snapshot of input for our proposed approach for user 1 consisting of keywords and genres. The Fig.6 shows top 20 books recommended to user1 based on keyword and genre similarity score.

	userId	original_ti	keywords	genres
0	1	The Blob	['junkie', 'heroin', 'prostitution', 'illegal', 'drugs', 'berlin', 'germany']	['Horror', 'Science', 'Fiction']
1	1	Force of E	['mutiny', 'ship', 'treasure', 'hunt', 'treasure', 'map', 'pirate', 'childrens', 'adve	['Drama', 'Action', 'Crime']
2	1	Os Norma	['treasure', 'map', 'magic', 'time', 'travel', 'titanic', 'cage', 'steampunk', 'mino	['Comedy']
3	1	Os Norma	['treasure', 'map', 'magic', 'time', 'travel', 'titanic', 'cage', 'steampunk', 'mino	['Comedy']
4	1	The Beach	['hippie', 'exotic', 'island', 'beach', 'map', 'group', 'dynamics', 'shark', 'attack',	['Drama', 'Adventure', 'Romance', 'Thriller']
5	1	Lara Croft:	['treasure', 'buddhist', 'monk', 'planetary', 'configuration', 'angkor', 'wat', 'ill	['Adventure', 'Fantasy', 'Action', 'Thriller']
6	1	Pusher	['riddle', 'treasure', 'treasure', 'hunt', 'archaeologist']	['Action', 'Crime', 'Drama', 'Thriller']
7	1	The Machi	['treasure', 'count', 'of', 'monte', 'christo', 'revenge', 'miniseries', 'woman', 'e	['Thriller', 'Drama']
8	1	Jeux inter	['sea', 'repayment', 'shark', 'attack', 'musical', 'gold', 'rush', 'treasure', 'hunt',	['Drama', 'History']
9	1	Pirates of	['exotic', 'island', 'blacksmith', 'east', 'india', 'trading', 'company', 'gold', 'mar	['Adventure', 'Fantasy', 'Action']

Fig.5. Snapshot of input showing movie with keywords and genre for user 1

The Passage	Maze of Moonlight	Skeleton Key	Wiseguy
Brain	Against the Odds	Sporting Chance	Atlas Shrugged
Stolen	The Beekeeper's Apprentice	Kingdom of Summer	Red
The Last Days	What Dreams May Come	Fight Club	The Wonderful Flight to the Mushroom Planet
Black	What Dreams May Come	Thrice Upon a Time	Red Dragon

Fig.6. Snapshot showing Top 5, 10, 15, 20 book recommendation for the user1

We conducted user study to form a testing dataset to calculate results. We conducted experiment on 150 users. Testing dataset included ratings provided by users for books and movie. We used movies ratings of users and performed experiment to get similarity score for books. After sorting books on the basis of score, we get top N books which can be recommended to users. We compared our results to data provided by user to calculate precision, recall and F-Measure.

From Table.4 it can be concluded that as the number of recommendations for a user increases from 5 to 10, precision of system decreases from 0.249 to 0.220 and recall of system increases from 0.268 to 0.367. There is a trade-off between precision and recall. However, F1-Measure of system increases from 0.258 to 0.275. Precision is the probability which shows how likely it is to be for a retrieved document to be relevant while Recall is the probability which shows how likely it is for a relevant document to be retrieved from the search. Recall of our system increases with number of recommendations.

Table.4. Prediction Performance of our approach for Top N recommendations for ω = 0.2

Top N	Precision	Recall	F-Measure
Top 5	0.249	0.268	0.258
Top 10	0.220	0.367	0.275
Top 15	0.199	0.457	0.272
Top 20	0.189	0.522	0.278

Table.5. Prediction Performance of our approach for Top N recommendations for ω = 0.4

Top N	Precision	Recall	F-Measure
Top 5	0.213	0.232	0.111
Top 10	0.210	0.382	0.135
Top 15	0.186	0.442	0.131
Top 20	0.149	0.501	0.14

Table.6. Prediction Performance of our approach for Top N recommendations for ω = 0.6

Top N	Precision	Recall	F-Measure
Top 5	0.232	0.272	0.125
Top 10	0.222	0.387	0.141
Top 15	0.201	0.410	0.134
Top 20	0.197	0.501	0.141

Table**.7.** Prediction Performance of our approach for Top N recommendations for ω = 0.8

Top N	Precision	Recall	F-Measure
Top 5	0.209	0.271	0.117
Top 10	0.198	0.342	0.125
Top 15	0.191	0.432	0.137
Top 20	0.180	0.502	0.132

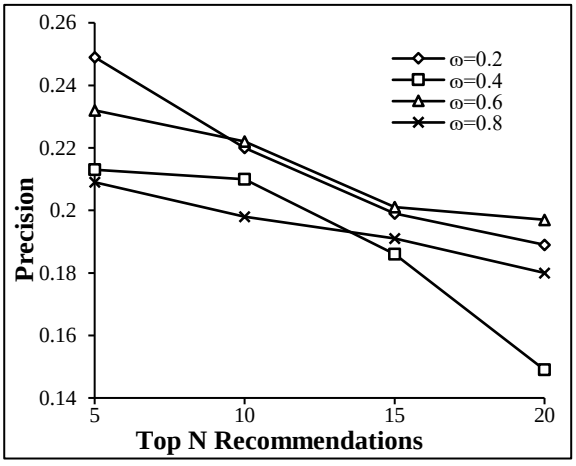

Fig.7. Graph showing Precision Comparison for Top N recommendations with different ω

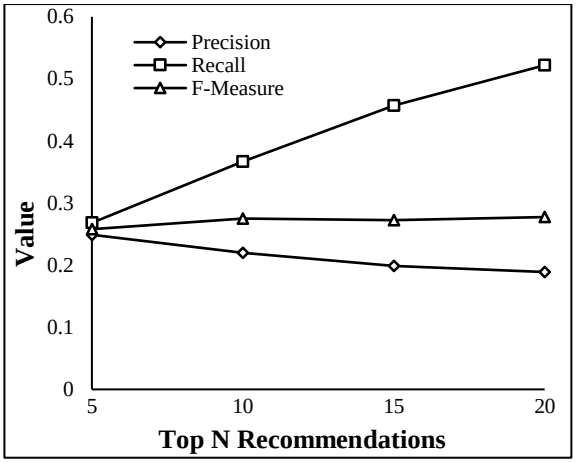

Fig.8. Graph showing Precision, Recall and F-Measure for Top N Recommendations

7. CONCUSIONS AND FUTURE SCOPE

In this paper, we have reflected upon the aspect of using keyword similarity and genre correlation for cross-domain recommender system. We used movies as source domain and books as target domain. We leveraged genre and keywords associated with books and movies to increase the accuracy of the system. We conducted a user study to experimentally verify our results that keywords and genres together can be used to improve the accuracy of the cross-domain recommender system. We can further explore more auxiliary data and find new algorithms for establishing similarity between the domains for cross domain recommendations.

REFERENCES

[1] B. Shapira, L. Rokach and S. Freilikhman, "Facebook Single and Cross Domain Data for Recommendation Systems", *User Modeling and User-Adapted Interaction*, Vol. 23, No. 2-3, pp. 211-247, 2013.

[2] A. Majid, L. Chen and G. Chen, "A Context-Aware Personalized Travel Recommendation System based on Geo-Tagged Social Media Data Mining", *International Journal of Geographical Information Science*, Vol. 27, No. 4, pp. 662-684, 2013.

[3] J.B. Schafer, D. Frankowski and J. Herlocker, "*Collaborative Filtering Recommender Systems*", Springer, 2007.

[4] B. Li, Q. Yang and X. Xue, "Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction", *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 2052-2057, 2009.

[5] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State of-the-Art and Possible Extensions", *Introduction to Data Mining and Knowledge Discovery*, Vol. 17, pp. 734-749, 2005.

[6] I. Cantador and P. Cremonesi, "Tutorial on Cross-Domain Recommender Systems", *Proceedings of 8th ACM Conference on Recommender Systems*, pp. 401-402, 2014.

[7] S.M. Choi and Y.S. Han, "A Content Recommendation System based on Category Correlations", *Proceedings of 5th International Multi-Conference on Computing in the Global Information Technology*, pp. 1257-1260, 2010.

[8] I. Fernandez-Tobias, I. Cantador, M. Kaminskas and F. Ricci, "A Generic Semantic-Based Framework for Cross-Domain Recommendation", *Proceedings of 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, pp. 25-32, 2011.

[9] S. Berkovsky, T. Kuflik and F. Ricci, "Mediation of User Models for Enhanced Personalization in Recommender Systems", *User Modeling and User-Adapted Interaction*, Vol. 18, No. 3, pp. 245-286, 2014.

[10] R. Lemdani, N. Bennacer, G. Polaillon and Y. Bourda, "A Collaborative and Semantic-based Approach for Recommender Systems", *Proceedings of 10th International Conference on Intelligent Systems Design and Applications*, pp. 469-476, 2010.

[11] Karamollah Bagheri Fard, Mehrbakhsh Nilashi and Naomie Salim, "Recommender System based on Semantic Similarity", *International Journal of Electrical and Computer Engineering*, Vol. 3, No. 6, pp. 751-759, 2013.

[12] Shampa Chakraverty and Mala Saraswat, "Review based Emotion Profiles for Cross Domain Recommendation", *Multimedia Tools and Applications*, Vol. 76, No. 24, pp. 1-24, 2017.

[13] Yang Rui, Wei Hu and Yuzhong Qu, "Using Semantic Technology to Improve Recommender Systems Based on Slope One", *Proceedings of International Conference on Semantic Web, and Web Science*, pp. 11-23, 2013.

[14] Magnus Sahlgren, "The Distributional Hypothesis", *Proceedings of International Conference on Linguistics*, pp. 1-18, 2006.

[15] Kaggle Datasets, Available at: https://www.kaggle.com/datasets, Accessed on 2017.

[16] C.N. Ziegler et al., "Improving Recommendation Lists through Topic Diversification", *Proceedings of ACM 14th International Conference on World Wide Web*, pp. 22-32, 2005.