[ICRTCET-2018]

ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

MULTI-DOMAIN RECOMMENDER SYSTEM

Kevin Jain[*1], Ojas Nadkar[2], Pranay Morye[3], Vivek Maurya[4] & Sowmiyaraksha R. N.[5]
[*1,2,3&4]Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute, Mumbai, India
[5]Assistant Professor, Department of Computer, Engineering and Information Technology, Veermata Jijabai Technological Institute, Mumbai, India

ABSTRACT

In recent times a lot of attention has been given to recommender systems and it is becoming the need of the hour to provide valuable information from the data that is growing exponentially. The underlying approach and algorithms hold the key to building an efficient recommender system. Most of the recommender systems work on a single domain and collaborative filtering performs well. But in the case of a multi-domain recommender system this approach cannot be extended, as it requires knowledge specific to only a single domain. In this paper, we aim to build a multi-domain recommender system to study the interests of users across different domains using Machine Learning principles. The thus produced recommendations improve upon a single domain recommender system and serve the needs on a wider platform. Common content can be used as a bridge between different domains and hence one can study the user's interest in one domain and make recommendations on the other domain. As per our findings, the closer an attribute is to the domain, the greater is the importance it gets when making recommendations. We observe that the best results are obtained when the attributes that define the domain more precisely are chosen and allocated more weight.

Keywords: *multi-domain recommender system, singular value decomposition, TF-IDF, content-based filtering, collaborative filtering.*

I. INTRODUCTION

Nowadays the amount of data that is produced is growing exponentially and it is quite difficult to process and find out useful information out of the myriad of options available. This is where recommender systems serve useful and cater to the needs of the users[8]. Systems which consider the user's taste in a specific domain and give out recommendations in the same domain like books, music, movies are termed as single domain recommender systems. A multi-domain recommender system addresses the problems faced in a single domain recommender system and tries to improve upon them. User's interest in a given domain can be studied and can be used to make recommendations in some other domain. This enables the system to make suggestions in the domains for which it does not require the user's taste in that domain. The user's interest can be understood based on his likings in some domain and two different domains can be linked using common features or content between them. Thus, making it possible to make recommendations in other domain from previously acquired knowledge in that domain.

Our aim is to build a system that can provide recommendations that span across domains of books and movies. We plan to implement the system by using collaborative filtering for intra domain recommendations[9] and content-based filtering for inter domain recommender system. Even though a user might be visiting a domain for the first time, their interests in the other domains can be used and reflected on this new domain as well and useful recommendations can be served.

The intra-domain collaborative filtering uses the singular value decomposition technique of matrix factorization on the user-item ratings matrix and predicts the user's ratings on unseen items. This method makes recommendations on a single domain only which could either be books or movies. For the inter-domain recommendations, collaborative approach does not serve useful and so a more content based approach has to be implemented. So the common attributes which can aptly and adequately define both the domains are selected. In our case for movies and books, the features selected are plot (storyline), genres, release date and the runtime. Different weights are adjusted to the different features using regression.

Experimental study consists of a comparison between different cases. It consists of different scenarios when only either of the features is considered for jumping domains or when all the four features are considered. Different cases by assigning optimized, equal or random weights to all the attributes is also tried and noted. The results are evaluated for accuracy and a comparison is drawn between them for analysis. This study enables us to draw a relationship between the domains and the features and how important each one is while making recommendations. So it serves quite useful for making conclusions.

II. RELATED WORK

Here we discuss the ongoing research and different approaches used to support multi-domain recommender systems. There is a considerable amount of research on the topic, but most of it is still a proof-of-concept. Due to the loose definition of a ‘domain’ it is very difficult to build systems which can make good recommendations over varied domains. Generally, some aspects of the domains chosen should be similar.

Matrix Factorization is useful when we want to make recommendations using item-user matrices. Singular Value Decomposition is an algorithm that decomposes the original matrix into the best lower-rank matrices. Therefore, it is an excellent method to recommend and compare using these lower ranked matrices [11].

When we want to recommend across multiple domains Collaborative Filtering is not enough. Therefore using a pre-filtering and post-filtering algorithm that makes use of content-based algorithms gives better results when it comes to cross-domain recommendations [1].

Collaborative Filtering used with content similarity measures give better results compared to only using Collaborative Filtering, though this can be done only when there are some similar attributes between the domains being considered[2].

The main problem with cross-domain recommendations arises when there is no user data. This is known as Cold Start. In such a case for the first few recommendations, data from related domains is used. This would not be a problem when there is data from the same domain which is trained.

When the modelling of user profiles is based on the preference of the user, we can give user specific recommendations. When these profiles are combined with collaborative filtering, it is observed that due to such profiling the accuracy of the cross domain recommendations is increased[4].

Identification and analysis on limitation of collaborative filtering based on users for multiple interests, results in a hybrid collaborative filtering method based on users and items. This hybrid method improves the accuracy of recommendations [5].

When we try and apply the generic recommendation system across domains, the simplest method is collaborative filtering. Here, a domain means one form of media such as movies, books, songs, etc[6].

Two methods most widely used in recommendation systems are Collaborative Filtering and Content-based filtering. We can try to get the best of both worlds by combining the two models to minimize the drawbacks of both and amplify the advantages. Several techniques of combining such as hybridization, switching, cascading, are found to increase the accuracy of recommendations [7].

This, paper combines various approaches to give the best recommendations for the movies and books domains.

III. PROPOSED METHODOLOGY

The proposed system consists of two databases: the user ratings database which consists of movies and book as separate tables and the metadata database which contains both movies and books in a single table. The user ratings database is used to find cosine similarity between item-pairs from both the tables. This similarity value is used as the output (y) in regression to determine weights. The metadata database is split into plot/summary, genre, length and release/publish date. All these parameters are quantified to give the input values (x1, x2, x3, x4) for regression. Therefore by breaking down the problem into inter-domain and intra-domain we obtain weights that give good recommendations across the movies and books domains.

Intra-domain approach
Here both the domains are considered independently and various algorithms are used to develop recommender systems in each domain. Collaborative Filtering which is used here gives excellent recommendations when it comes to a single domain.

Matrix factorization using singular value decomposition (SVD)
There are many ways to decompose and factorize a matrix but SVD is quite useful for making recommendations. The user-ratings matrix is the original data that we have and this matrix is broken down into 3 matrices which are an accurate approximation of the original matrix. Mathematically it decomposes ***R*** into two.
R = user ratings matrix U = user features matrix
∑ = diagonal matrix of singular values (essentially weights)
VT = movie features matrix
"U" matrix is the user features matrix which represents how much the users like each feature and the "VT" matrix is the movie features matrix which represents how relevant is each feature to each movie. This helps us derive the underlying tastes and the preferences of the users from the raw data.

This technique of low rank factorization is quite accurate and scales well to large datasets as well. It outperforms other techniques for collaborative filtering and yields meaningful results as well. It is capable of computing large data at a much faster rate and is quite efficient compared to other methods.

In this approach we use combine the data of both the domains and with the help of several algorithms obtain the input values for regression. This approach uses content based filtering approach, and the features that are common to both movies and books are selected to make recommendations across the two domains. We use the plot/summary of movies/books, genre, release/publish date and length.

Plot similarities using TF-IDF (Term-frequency inverse document frequency)
TF-IDF weight is used as a statistical measure of how important a word is to a document in a collection. This is a product of two terms: the term-frequency(TF) and the inverse document frequency(IDF)[13]

TF-measures how frequently a term appears in a document.
TF(t) = (number of times term t appears in the document) / (total number of terms in the document) *IDF(t)* = log[(total number of documents) / (number of documents with term t in it)]
Using this algorithm certain keywords are obtained with a coefficient depending on how strongly they define the plot. The similarities are mapped based on the keywords and similar movies/books pairs are formed of the top results.

Unique genres are identified from the datasets of movies and books and each of them is mapped into a set of genres which are common to both movies and books. After this mapping, similarities are drawn for the respective movies/books based on how close the genres are to each other. These similarities are based on the results of a research paper which clusters the genres based on their similarity[10].

The time of release of the respective movies and books is also considered as it reflects the trends and the tastes of those times. The release is in the date format and using the year of release it is converted into a coefficient which determines the movies/books from new to old. The normalization is done using a formula:

$$TF\text{-}IDF\ weight = \mathrm{TF}(t) * \mathrm{IDF}(t) \qquad (2)$$

$$RD = 1 - \frac{max - year\ of\ release}{max - min} \qquad (3)$$

Where *max=2017*, *min=1900*

So, more recent the movie, the value of coefficient is closer to 1.

Length

The length of the movies or the number of pages in a book can also be considered a measure if a user might like it or not. Someone might just prefer short movies or someone might love to read long books. So the books and movies are distributed across three categories of short, medium and long. Short movies would range from 0-40 mins, medium ones would range from 40-120 mins and the long ones above 120 mins. Similarly ranges are defined for books and they are categorized. Similarity can then be drawn from this feature as well.

All the features link the two domains but not equally. Each of them affects the recommendations to a different extent. So to find out how important each one is and their impact on the recommendations, regression is used to find out the weights associated with each feature.

Regression

Now that we have our features finalized all we need to do is find out the weights (indicates how important each feature is while making a prediction) that should be assigned to them and then output the results. This is one of the last yet major steps involved in making accurate predictions.

For this purpose, we compare a movie pair and find out the similarity based on the ratings given to it by multiple users. We use Pearson correlation coefficient in this case. Using this we get similarities between a movie and the rest of the movies in the database. We treat this as the output Y that we are going to use in our regression function[14].

1) Computation of Y

The output vector *Y* which is considered to be the actual similarity is computed for two movies *i* and *j* as shown:

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \overline{R_i}) * (R_{u,j} - \overline{R_j})}{\sqrt{\sum_{u \in U}(R_{u,i} - \overline{R_i})^2}\sqrt{\sum_{u \in U}(R_{u,j} - \overline{R_j})^2}} \qquad (4)$$

Here, *R* is the ratings matrix of Users and Movies, and *U* is the set of users who have rated movies.

We treat the four features as the four inputs *x1*, *x2*, *x3*, *x4* to train the weights and set them to get the optimal results. Similarities among the movies/books based on those features have been individually explained above.

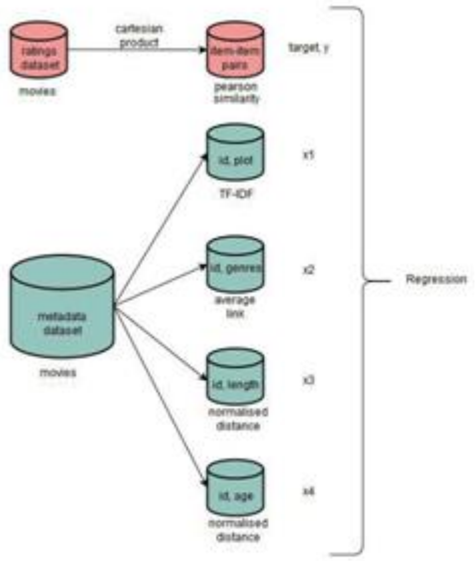

Fig.1. High-level architecture multi-domain recommendation (books & movies)

IV. EXPERIMENTAL SETUP AND RESULTS

This section presents the implementation of the proposed methodology, and graphs are included for better visual representation. We have four input values for which we need to find weights which give the best recommendations. We compare two parameters: Mean Square Error (MSE) and R^2 score. By using different combinations of weight vectors we observe that the least MSE value and the greatest R^2 score is obtained for the weights obtained from regression, hence proving that the best recommendations are obtained by using these weights.

A. Comparison

Using the above mentioned methods different combinations of weights were assigned to each attribute for experimentation purposes and the results were analyzed. We found out that the plots which define the story of the movie/book in a gist hold the utmost relevance while making recommendations followed by genre, release date and run-time respectively. This certainly makes a clear point that the features which describe the movie/book more aptly are given more weightage while the other attributes may affect the recommendation but to a lesser extent. The more the number of attributes and the more relevant the features selected, it is sure to have a positive impact on the outcome and more refined recommendations can be obtained.

Table.I represents the summary of the various experiments done to study the relationship between the feature and its relevance to the recommendations. It is compared based on measures like the MSE score and the R^2 score. A lower MSE score or a higher R^2 score means that the combination of weights is better.[15]

TABLE I. Weight matrix combinations comparison with MSE and R^2 Score

Sr No	Features	Weights	MSE	R^2 Score
1	Everything with optimised weights	[0.97093111, 0.00722658, 0.06556827, -0.00121337]	0.00802	0.56898

ISSN 2348 – 8034
Impact Factor- 5.070

2	Only Tf-idf	[1,0,0,0]	0.01097	0.41073
3	Only genre	[0,1,0,0]	0.44074	-22.68554
4	Only release date	[0,0,1,0]	0.51408	-26.62686
5	Only run-time	[0,0,0,1]	0.65777	-34.34878
6	Everything with equal weights	[0.25,0.25,0.25 ,0.25]	0.26689	-13.34287

B. Mean Square Error (MSE)
MSE is the estimate of variance of residuals, or non-fit, in the population[12]

$$MSE = \frac{SSE}{n-m} \qquad (5)$$

where *n* is the sample size and *m* is the number of parameters in the model (including intercept, if any).

C. R^2 Score
R^2 is a standardized measure of degree of predictively, or fit, in the sample where n is the sample size and m is the number of parameters in the model (including intercept, if any)[12].

$$R^2 = 1 - \frac{SSE}{SST} \qquad (6)$$

where *SSE* is the sum of squared error (residuals or deviations from the regression line) and *SST* is the sum of squared deviations from the dependent's *Y* mean.

D. Relation between R^2 Score and MSE [12]

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-m} = 1 - \frac{SSE/(n-m)}{SST/(n-1)} = 1 - \frac{MSE}{\sigma^2_y} \qquad (7)$$

The following Figure.2 and Figure.3 shows a comparison of the attributes with respect to MSE and R^2 Score

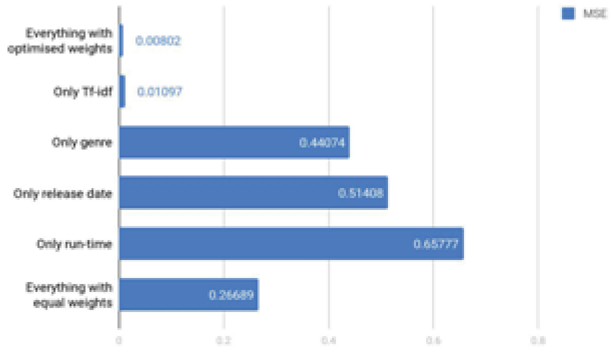

Fig.2. MSE variation with weight values

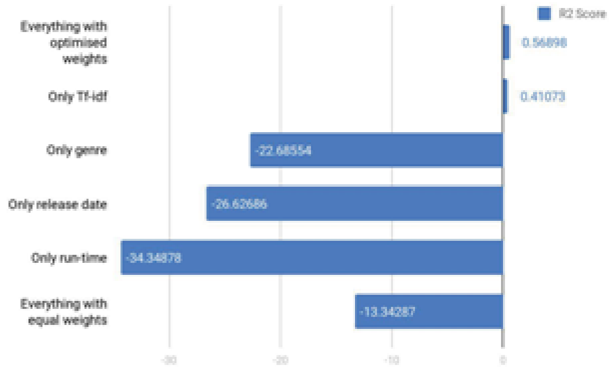

Fig.3. R^2 Score variation with weight values

V. CONCLUSION

Recommendations are a very important part of the Internet and serve as a useful tool to provide relevant information at a glance. From search engines to video, movie, books recommenders, the variety is vast. These recommender systems provide users with options to make better decisions and in turn provide user data to the system. The need for multiple-domain recommender systems is ever increasing. Our paper makes use of various algorithms and approaches to successfully recommend across movies and books. We observe that for multi domain recommendations, a more content based approach is necessary and yields better results compared to other methods. Furthermore, this system can be expanded to include more domains like music, TV shows, etc. Even context aware trends like the live location, time, mood can be included to provide more refined and accurate recommendations to every user.

REFERENCES

1. *Prota, "Context-Aware Technique for Cross-Domain Recommender Systems", IEEE Brazilian Conference on Intelligent systems, 2015.*
2. *Paolo Cremonesi, Antonio Tripodi, Roberto Turrin, "Cross-Domain Recommender Systems", IEEE International Conference on Data Mining Workshops (ICDMW), 2011*
3. *Ignacio Fernandez-Tobias, Paolo Tomeo, Iván Cantador, Tomasso Di Noia, Eugenio Di Sciascio, "Accuracy and Diversity in Cross-domain Recommendations for Cold-star Users with Positive-only Feedback", 10th ACM Conference - Boston, Massachusetts, USA, 2016*
4. *Malivardo Rodrigues, Gabriela O. Mota da Silva, Frederico Araujo Durao, "User Models Development based on Cross-Domain for Recommender Systems", ACM Press the 22nd Brazilian Symposium, 2006*

5. *Yu Li, Liu Lu, Li Xuefeng. "A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce", Elsevier, Expert Systems with Applications, 2005*
6. *Pinata Winoto, Tiffany Tang, "If You Like the Devil Wears Prada the Book, Will You also Enjoy the Devil Wears Prada the Movie? A Study of Cross-Domain Recommendations", Springer, 2008*
7. *George Lekakos, Petros Caravelas, "A hybrid approach for movie recommendation", Springer, 2006*
8. *Recommender Systems https://yanirseroussi.com/2015/10/02/the-wonderful-wo rld-of-recommender-systems/*
9. *Collaborative Filtering in Recommender Systems - http://recommender-systems.org/collaborative-filtering/*
10. *Genres Distance Matrix - https://www.scribd.com/document/72221993/Analysis-and-Clustering-of-Movie-Genres*
11. *Matrix Factorization for recommendations - https://beckernick.github.io/matrix-factorization-recommender/*
12. *Comparison between MSE and R2 Score - https://stats.stackexchange.com/questions/32596/what-i s-the-difference-between-coefficient-of-determination-a nd-mean-squared*
13. *TF-IDF - http://www.tfidf.com/*
14. *Regression - https://machinelearningmastery.com/linear-regression-f or-machine-learning/*
15. *http://scikit-learn.org/stable/modules/generated/sklearn. metrics.r2_score.html.*