Open Access Library Journal
2020, Volume 7, e6483
ISSN Online: 2333-9721
ISSN Print: 2333-9705

A Novel and Hybrid Approach of an Indian Demographic Movie Recommender System

Ananth G S[1], K. Raghuveer[2], Dayananda R[3], Kashyap R[1]

[1]Department of MCA, National Institute of Engineering (NIE), Mysuru, India
[2]Department of ISE, National Institute of Engineering (NIE), Mysuru, India
[3]CEO, Kitchen 365, Mysuru, India
Email: ananth.gouri@nie.ac.in

How to cite this paper: Ananth G S, Raghuveer, K., Dayananda R and Kashyap R (2020) A Novel and Hybrid Approach of an Indian Demographic Movie Recommender System. *Open Access Library Journal*, **7**: e6483.
https://doi.org/10.4236/oalib.1106483

Received: May 29, 2020
Accepted: July 17, 2020
Published: July 20, 2020

Abstract

India is a demographic democratic country having a population of nearing 140 crores and with different people of various religions, communicating numerous languages, wearing different varieties of clothes. India is also a cacophony of languages, with more than 1500 films being produced every year in its 20+ languages. Recommender systems give personalized outputs in the form of the information being processed. But unfortunately, there is very little personalization done or the data available for this voluminous demographic attribute possessed by India. For example, though there are different platforms like Amazon prime videos, Netflix, tickets booked through www.bookmyshow.com/ to watch movies but not restricted to just Hindi and English (the two official languages of India)—there is little concentration towards the demographic data of Indian languages. In this paper, we present a novel way of creating an Indian Demographic Movie Recommender System (IDMRS) making full utilization of the various demographic attributes available. IDMRS is a system capable of filtering and providing personalization to users in five regional south Indian languages. This system makes use of various characteristics and demographic attributes, such as age, gender and occupational details for the generation of recommendations. Also, a curated dataset, similar to MovieLens dataset, is evolved with this system and is evaluated with various performance metrics.

Subject Areas

Information Retrieval

Keywords

Demographic Filtering (DF), Information Retrieval (IR), Recommender Systems (RS), Similarity Index (SI)

1. Introduction

With this outburst of data in recent years, it's a tough task for a user to select and pick a nice movie for watching. Also, with the huge options and preference choice made easily available, selection of data in a particular domain is not at all easy. Recommender systems are used to provide new personalization and also to showcase existing recommendations. These recommender systems can be used for recommendations on various domains like books, songs, jokes, news, online products and not limited to movies.

Research and literature survey suggest various approaches to creating an RS. Also, it is important to note that an RS has its own positives and negatives and we should be careful while selecting a suitable approach for implementation.

The most commonly used approaches of RS are via Collaborative Filtering methods and Content-based filtering techniques. IDMRS makes use of a logical combination of both these approaches and hence is hybrid in nature.

Till now, there are not many details available about demographic personalization for the movie domain except [1]. IDMRS makes use of recommending personalization to a user in five regional languages: Kannada, Hindi, Telugu, Tamil and Malayalam. This way we were able to concentrate mainly on the South Indian dialect languages.

IDMRS could group users based on attributes like age, gender. The personalization produced by IDMRS could determine the user's need for a movie recommendation. A movie recommendation was based on the similar movie taste given as an input or a completely new movie. The same was achievable for a new user as well. A new user could get a movie recommendation based on previous interests and preferences or a completely new movie as a recommendation. This way the common cold-start problem faced in RS was avoided to a certain extent. To be precise, a new user lacks good recommendations due to lack of enough ratings. IDMRS utilizes the user demographic data to avoid cold-start problems.

IDMRS was able to customize movie name generations. That is, the demographic information available can be used to provide customized personalization as per the language of choice.

1.1. Motivation

One of the most common problems of available movie data sources like Netflix, Amazon Prime, and now the new Jio Movies, Airtel Movies, Flipkart videos is the lack of a comprehensive list of Indian movies.

Also, it is unfortunate to note that except Netflix, these data sources do not provide their dataset for research and future learnings.

1.2. Contributions

Through IDMRS,

- A web portal for the curated dataset collection of data: A web portal where a user signs up by filling demographic data. Then the user can provide a like or a dislike as a movie rating.

- Indian Demographic Movie Dataset: With the implementation of IDMRS, we were able to create a dataset of Indian Demographic Region and especially for the South Indian dialect languages.
- Analysis of the data: We made use of a logical combination of content and collaborative techniques for filtering and analysis of this regional dataset.

2. Related Work

In [2] an Indian regional movie dataset the very first of its kind-database of regional Indian movies, users and their ratings is created. It consists of movies from 18 regional languages, capturing ratings from 919 users and for 2851 movies.

The analysis of this dataset is done using some supervised and unsupervised collaborative filtering techniques like Probabilistic Matrix Factorization, Matrix Completion, and Blind Compressed Sensing etc.

In [3] the researchers make use of the MovieLens dataset [4] and try to exploit the user demographic attributes for solving the cold-start problem in the Recommender System. The paper is structured into overall five phases where in the last phase they also provide directions for future research after the MovieLens dataset is evaluated.

Also, this paper suggests a novel framework to solve the cold-start problem by the utilization and influence of demographic attributes on user ratings and also to assist the RS to improve its recommendations.

3. Proposed Approach and Work

The IDMRS recommendation generation for a user works at five stages:

I. Data collection
II. Similarity index calculation
III. Recommendation generation
IV. Recommendation evaluation
V. Presentation to users or regeneration of recommendations.

For IDMRS, the similarity index calculation and recommendation generation are both demonstrated in a single phase.

The workflow of IDMRS is shown in **Figure 1**.

Table 1 is the description of user demographic data.

Each user would need to register and login into the site before able to see recommendations or even before rating a particular movie. This also made sure that a session was maintained all the time till logout. The IDMRS also can be considered as a session-based RS.

Table 2 shows the description of tables in the IDMRS.

3.1. Metadata Information

The IDMRS dataset was constructed in the Phase-1: Collection of data consisted of majorly three tables: User, Movie and Ratings as in **Table 2**. The user description

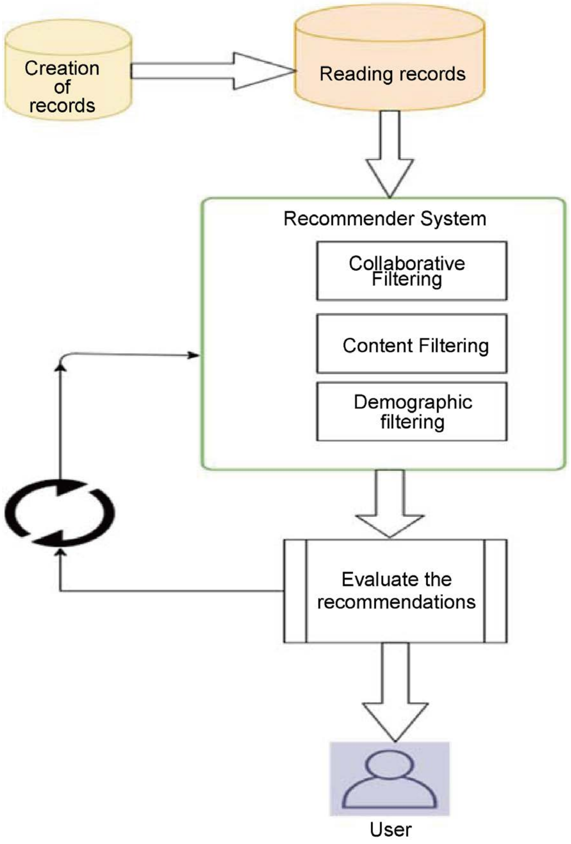

Figure 1. The workflow of IDMRS.

Table 1. User demographic data in encrypted format.

UserId	Name	Gender	Occupation	Email	Age	Mobile
01	Kashyap	M	Student	k***yapr@gmail.com	24	99854**763
02	Namitha	F	Lawyer	nam***@gmail.com	28	98745**321

Table 2. Description of tables in IDMRS.

Sl. No.	Table Name	Description
1	User	This is the user table consisting of user demographic data
2	Movie	This is the movie table where all of the movie related information is stored
3	Ratings	The ratings of movies for each individual film is stored in this table.

is described in Table 1.

Table 3 describes the Movie table and Table 4 describes the Ratings table.

Table 5 gives the overall picture of the curated dataset.

3.2. Calculation of Similarity Index and Generation of Recommendations

First of all, the curated dataset had a train/test split ratio of 8:2. In our work of

Table 3. Desc of movie table.

Sl. No.	Table field	Example field value
1	Movie_Id	231
2	Movie_Name	Bharat Ane Nenu
3	Movie_Language	Telugu
4	Movie_Desc	Description
5	Movie_duration	Running time of the movie

Table 4. Desc of ratings table.

Sl. No.	Table field	Example field value
1	UserId	27
2	Movie_Id	231
3	Ratings_stars	1 to 5 (High: 5)
4	Likes	Boolean 0 or Boolean 1

Table 5. Curated dataset details.

Sl. No.	Table details	Total count
1	Users	266
2	Movies	367
3	Languages	5
4	Ratings	7500

calculating the similarity index and generation of recommendations, we created a Hybrid Recommender for our approach.

A hybrid recommender is one which makes use of logical combinations of both content filtering and collaborative filtering techniques. Hybridization techniques are further classified as monolithic design, parallelized design and pipelined design of hybridization [5].

Our approach makes use of a monolithic hybrid design model where in for the:

Input: User Id and Movie name

Output: Similar movie based on the input movie name and sorted on the basis of expected ratings by that particular user

Also, to make sure that we had our hybrid recommender design working, we split our implementation to work on and see the results of these recommenders:

1. Popularity recommender
2. User based recommender
3. Item based recommender

For the popularity recommender, we used Top-N functionality, where in N is most popular set of films as per the count of user ratings sorted from high to low.

In the case of User based recommender, cosine similarity index was used. The similarity index is defined as in (1)

$$\cos\theta = \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|\cdot\|\boldsymbol{b}\|} \tag{1}$$

Hamming distance was used for similarity in the case of Item based recommender system. One of the reasons for using Hamming distance was, we used movies of same language and length or duration greater than 2 hours as a feature vector for item based recommender system and in presented binary format.

The hamming distance between 2 movies say A and B with "i" as the feature vector is as defined in (2)

$$\sum\left|A_i - B_i\right| \tag{2}$$

The recommendations of movies with highest value of cosine similarity to a value of 1 and the least hamming score were generated.

Table 6 depicts an example of the hamming distance calculated between 2 movies of the same length:

Note that, the hamming distance of unequal lengths is always 1 and equal lengths is a binary 0 value. All such movies where in the hamming distance was 1 was calculated and grouped together.

The hybrid recommender made a logical integration of user based recommender with item based recommender.

3.3. Evaluation of the Recommendations

Common Modelling Metrics Used in Recommender Systems

The evaluation metrics used in our approach is Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). They are the most common modelling metrics used in the case of Recommender Systems [6].

MAE and RMSE are defined as in (3) and (4)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - y_i}{\sigma_i}\right)^2 \tag{3}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - y_i}{\sigma_i}\right)^2} \tag{4}$$

Table 7 shows the MAE and RMSE evaluation modelling metrics of the IDMRS.

The IDMRS was able to generate recommendations by also utilizing the demographic attributes and the metrics had a fair amount of improvement.

Table 6. Hamming distance for movie with duration as a feature vector.

Sl. No.	Movie Title	Movie Duration (in minutes)	Hamming Distance
1	ABCD-1	143	1
2	ABCD-2	154	

Table 7. Evaluation of IDMRS models.

Sl. No.	Model type	Type variant	Evaluation Metric	
			MAE	RMSE
1	Non-personalized	Popular RS	0.83	0.97
		User-based RS	0.76	0.87
2	Personalized	Item-based RS	0.48	0.92
		Hybrid RS	0.73	0.45

Table 8 displays the demographic attributes used for better recommendations.

3.4. Presentation of Recommendations to Users

Though the complete IDMRS site was developed using the open source PHP framework, Python language code was used on the dataset to obtain recommendations.

We had to execute Python scripts within PHP and display the outputs back onto PHP. We came across [7] to solve this issue. The escapeshellcmd and shell_exec() functions were used to call Python within PHP.

4. Results and Discussion

In this section, we run our model for Precision, Recall and F-measure score of the generated recommendation list.

Precision, also called the positive predicted value, as in this case of IDMRS is the fraction of relevant movies predicted by the system to the overall retrieved movies. A movie is relevant when its rating is greater than or equal to 3.8 or else it is irrelevant and this is fetched from the curated dataset.

Precision is defined as in (5)

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{5}$$

Likewise, recall is the number of related movies predicted by the model to the total number of relevant movies.

A recommended movie is also the predicted movie whose rating is again greater than or equal to 3.8 and this is the output or the result of the prediction algorithm.

Recall is also called sensitivity. It is defined for our IDMRS as in (6)

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{6}$$

Finally, the mean between the precision and recall is measured by a score called F-Measure. When the measure is having a value of 1—then it means that, we are giving equal significance to both precision and recall.

Now F-Measure becomes F-1 (as in equality to a value of 1 to both precision and recall).

F-1 measure to our IDMRS is derived as in (7)

Table 8. Demographic attributes used for IDMRS improved recommendations.

Attribute	Value	Count	Total	Avg. MAE	Avg. RMSE
Gender	Female	58	266	0.84	1.34
	Male	208		0.96	
Occupation	Student	67		0.77	1.38
	Lawyer	24		0.49	
	Engineer	175		0.67	
Age	<18 years	26		0.44	1.14
	18 to 35 years	78		0.48	
	35 to 50 years	84		0.21	
	>50 years	78		0.19	

$$F_\beta = \left(1+\beta^2\right)\times\left(\text{Num}/\text{Den}\right) \tag{7}$$

where,

$$\text{Num} = \text{Precision}\times\text{Recall}$$

$$\text{Den} = \left(\beta^2\times\text{Precision}\right)+\text{Recall}$$

$$\beta = 1$$

Table 9 depicts the precision, recall and F-1 scores for the user with UserId-5 for the IDMRS hybrid recommendation output. The Trial I and Trial II details are depicted as per Table 10(a) and Table 10(b), through the confusion matrix. The train to test split was an 8:2 ratio.

For the few trials done, for our model of IDMRS, we were able to achieve a precision of nearly 50% and recall of more than 50%. We are able to achieve an F1-score of 54%. The F1-score denotes the performance of the Recommender System. The higher the value, the better it is.

One of the observations made during the run of IDMRS was, with higher number of trials and an aggregated average of Precision and Recall taken, the performance of the RS could improve drastically.

Also with the addition of more number of demographic attributes and fine tuning of the same, the F1 score could raise by at least another 10%. These attributes observed are could be the movie maker characteristics, the genre or the type of the movie.

Unfortunately when the IDMRS dataset size is compared with the other existing movie datasets, the size of IDMRS is very small and tiny. Adding more attributes and increasing the size of the dataset is discussed in challenges.

5. Challenges

These are a few of the challenges faced while working with our work:

- The total number of Indian languages is 22 in [8]. The collection of data for these 22 languages is indeed a herculean task.

Table 9. Precision, recall and F1 score.

Sl. No.	Trials	Precision	Recall	F1 Score
1	Trial-I	0.45	0.55	0.495
2	Trial-II	0.54	0.6	0.568

Table 10. Confusion matrix with TP, TN, FP & FN values for 2 trials. (a) The trial-I values; (b) The trial-II values.

(a)

	+ve	−ve
+ve	TP 10	FP 12
−ve	FN 8	TN 5

(b)

	+ve	−ve
+ve	TP 12	FP 10
−ve	FN 5	TN 8

- The data collected for the 5 languages of our implementation is only for a period of 5 years. If all of the year's data needs to be collected, then it's not an easy task and is indeed a difficult challenge.
- We should understand the restriction of data and the huge sizes of movies in different formats. The dataset of IDMRS only stored the metadata of movies. Storing the physical movies is not easy and may not be needed as well.
- The attributes used for producing demographic outputs are few in number. If the dataset has more number of fields as demographic data, then procuring recommendations out of them has many numbers of permutations and combinations.

6. Conclusions and Future Work

In this work we provided a novel approach of evaluating the demographic attributes available in the curated dataset to provide and recommend movies to new users. The movies were sometimes the popular ones recommended and few times absolute new users were provided with interesting movies; thus the problem of cold-start was also avoided to an extent.

The IDMRS can be also considered as a session-based RS. All the operations of the RS had to be done with the user logged-in.

Further work and research can be done using the dataset of IDMRS. More number of options for obtaining hybrid recommendations can be made if there are a numerous number of demographic attributes, which can be added to the curated dataset through web scraping or any other feature.

Finally, the same approach of working with movies as in the case of IDMRS can be applied to other domains as well. IDMRS can work as a generic platform.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Indian Population in 2020. https://countrymeters.info/en/India

[2] Indian Regional Movie Dataset for Recommender Systems. https://arxiv.org/abs/1801.02203

[3] Exploiting User Demographic Attributes for Solving Cold-Start Problem in Recommender System. https://www.researchgate.net/publication/272908674_Exploiting_User_Demographic_Attributes_fo r_Solving_Cold-Start_Problem_in_Recommender_System

[4] The MovieLens Research Website. https://grouplens.org/datasets/movielens/

[5] Jannach, D. (2010) Hybrid Recommender Approaches. Recommender Systems: An Introduction. Cambridge University Press, Cambridge.

[6] Herlocker, J. (2004) Evaluating CF Recommender Systems. *ACM Transactions on Information Systems*, **22**. https://doi.org/10.1145/963770.963772

[7] Calling Python Files within PHP. https://www.tutorialspoint.com/How-to-call-Python-file-from-within-PHP

[8] Over 22+ Indian Languages. https://www.traveldudes.org/travel-tips/india-country-over-22-languages/9384