MOVIE RECOMMENDATION SYSTEM BASED ON CONTEXTUAL USER INFORMATION

Shruti Shelke and Snehal Prabhu
University of Massachusetts Amherst
sshelke@cs.umass.edu,scprabhu@umass.edu

ABSTRACT

In our research project, we propose to build a movie recommendation model that also considers some of the the contextual information about the user such as their dominant emotion and end emotion while watching the movie, number of interactions with the movie, as well as their geological location, weather at the location when the movie was watched, mood, and time of day. We also use the movie genres, directors, cast and time of release to supplement the model. We believe that these factors affect the user's opinion of a movie, and hence the ratings that the user will give it. We use decision tree and random forest approaches to build our models. We also perform evaluations to compare the quality of the results obtained from our models with the metrics mean absolute error and mean squared error.

1 INTRODUCTION

Previous research on movie recommendation models have either been content based, employ collaborative filtering or are geo-location based. In our research, we will build a movie recommendation model that incorporates both the demographic and contextual information of the user as well their geographic location. We also intend to compare the recommendation model that uses a dataset with contextual user information against a model that uses variables only appearing in the regular Movie Lens dataset, i.e, no contextual or demographic information of the user besides age and gender. This contextual information is defined as 'information that describes the situation in which the user consumed the item'. We believe that movie recommendation can be improved if we study the contextual information of the user in addition to their age and gender such as emotions, location and other demographic information since these factors affect a user's preference for watching movies. We can also employ this recommendation model to other use cases such as music recommendation, online shopping, restaurant hunting, and many more that are also affected by the personal characteristics of the user.

We want to study the effect of a movie recommender system if we add more user based parameters for recommendation. The research question we want to address is whether incorporating the contextual information of users in a movie recommender model actually improves the results. We built two models to test our claims on our dataset, which were decision tree model and random forest model. Since our dataset was relatively small, about 2300 entries, and highly sparse with thirty variables, we chose these models because it would be less fruitful to run it on other supervised and unsupervised machine learning algorithms such as neural networks or clustering.

2 DATASET

We use the LDOS-CoMoDa dataset for our project. It is an extensive movie recommender dataset with thirty variables, twelve out of which are contributors of the contextual information we seek to address. In addition to ratings of movies, this dataset has a dozen other attributes that describe the user movie watching experience in which the movie was rated. For example: emotional response to the movie, physical conditions of the user such as healthy or ill, number of times the user has watched the movie, user's mood and many more. There are 2296 total ratings made by 120 users from 6 different countries. We intend to divide our data set into 80 percent training and 20 percent testing randomly. The user-based contextual variables are outlined in the table below. Each variable is a categorical variable, with each level represented numerically in the model.

No	Type	Possible values
1	time	Morning, Afternoon, Evening, Night
2	daytype	Working day, Weekend, Holiday
3	season	Spring, Summer, Autumn, Winter
4	location	Home, Public place, Friend's house
5	weather	Sunny / clear, Rainy, Stormy, Snowy, Cloudy
6	social	Alone, My partner, Friends, Colleagues, Parents, Public, My family
7	endEmo	Sad, Happy, Scared, Surprised, Angry, Disgusted, Neutral
8	dominantEmo	Sad, Happy, Scared, Surprised, Angry, Disgusted, Neutral
9	mood	Positive, Neutral, Negative
10	physical	Healthy, Ill
11	decision	User decided which movie to watch, User was given a movie
12	interaction	first interaction with a movie, n-th interaction with a movie

Our goal is to use more user variables in addition to the demographic variables used in the MovieLens data set, which are age, gender, occupation and zipcode and test whether the addition of these variables makes a difference to the performance of our model. Accordingly, we divide our dataset into four subsets to run on our models. Our four subsets of the data are influenced by the paper 'Predicting and Detecting the Relevant Contextual Information in

a Movie-Recommender System', co-authored by Professor Andrej Kosir from the University of Slovenia who curated the dataset. In this paper, the authors analyze the contextual variables of the LDOS Comoda Dataset and curate a smaller list of those variables that are the most important to a movie recommendation system. The authors state that using only relevant variables for a recommendation system not only improves the quality and efficiency of the model, but also that using irrelevant variables could even decrease the performance of the model due to the addition of noise. Hence, it is necessary for us to narrow down the variables that make actual significant contributions to our recommender system. In the paper, the researchers use statistical tests and power analysis to detect whether a given contextual variable is important for making predictions. This testing is done with the help of the Pearson's 2 test and the Freeman–Halton test. Upon conducting these tests, they narrow down six important contextual variables of interest.

The six contextual variables deemed relevant to the model are: *'endEmo', 'dominantEmo', 'mood', 'physical', 'decision', and 'interaction'*. Furthermore, *'location' and 'daytype'* were found to be inconclusive, and *'time', 'season', 'weather', 'social'* were deemed irrelevant. After variable selection of the contextual parameters, several recommender system models based on collaborative filtering are also run on subsets of these variables. Observations of the root MSE outputs corroborate the findings that the relevant contextual variables improve the performance of the model while the irrelevant variables hinder efficiency and effectiveness. Based on these conclusions, we run our recommender models on the following four subsets of variables.

Name	Type	Parameters
Subset 1	All User Data	'userID', 'itemID', 'age', 'sex', 'city', 'country', 'time', 'daytype', 'season', 'location', 'weather', 'social', 'endEmo', 'dominantEmo', 'mood', 'physical', 'decision', 'interaction', 'director', 'movieCountry', 'movieLanguage', 'movieYear', 'genre1', 'genre2', 'genre3', 'actor1', 'actor2', 'actor3', 'budget', 'rating'
Subset 2	No Additional User Data (MovieLens variables)	'userID', 'itemID', 'age', 'sex', 'city', 'country', 'director', 'movieCountry', 'movieLanguage', 'movieYear', 'genre1', 'genre2', 'genre3', 'actor1', 'actor2', 'actor3', 'budget', 'rating'
Subset 3	Important Contextual Data	'userID', 'itemID', 'age', 'sex', 'city', 'country', 'endEmo', 'dominantEmo', 'mood', 'physical', 'decision', 'interaction', 'director', 'movieCountry', 'movieLanguage', 'movieYear', 'genre1', 'genre2', 'genre3', 'actor1', 'actor2', 'actor3', 'budget', 'rating'
Subset 4	Important and Inconclusive Contextual Data	'userID', 'itemID', 'age', 'sex', 'city', 'country', 'time', 'daytype', 'location', 'endEmo', 'dominantEmo', 'mood', 'physical', 'decision', 'interaction', 'director', 'movieCountry', 'movieLanguage', 'movieYear', 'genre1', 'genre2', 'genre3', 'actor1', 'actor2', 'actor3', 'budget', 'rating'

Figure 1: Subsets of parameters used to test

Subset 1 is created so that we can analyze the dataset as a whole. Subset 2 is created to analyze the repercussions of having no contextual or demographic information other than the ones used by the Movie Lens dataset. For Subset 3, want to test whether these 'important' contextual variables do actually improve performance of our model. Lastly, with Subset 4, We want to test whether these 'inconclusive' contextual variables affect the performance of our model.

3 RELATED WORK

To make movie recommendations more personalised for the users a recommendation model can take into account user's profile to match user preferences. User's profile may contain user's personal information such as age, gender, demographics as well as ratings for movies and preferences. These parameters are used to estimate ratings for unseen movies and improve the recommendations.

The paper "A location-based movie recommender system using collaborative filtering" talks about the importance of location as a factor for a movie recommender system using Collaborative Filtering and the Pearson Correlation Coefficient. They claim that by adding the location of user in the user profile for selecting peers with the similarity function (Figure 2) and for item recommendations the system can generate better recommendations for the user. The Pearson Correlation function calculates the similarity between the users u and v, using the set of movies I, both user have watched. Where r_{ui} and r_{vi} are the ratings of movie i by users u and v respectively and r_u and r_v are the mean ratings by users u and v respectively. After we get the similarity score of the users we select the top similar users to the active user by a predefined thresh-hold value. While calculating the predictions of ratings for the active user, the users which share the same geographic location as the active user are given higher preference by adding a constant term alpha to the similarity score thus increasing their score. The ratings by users with a similarity score exceeding the thresh-hold are used for predicting the ratings for the unseen movies. The pred(u,i) function (Figure 3) uses the similarity score, the ratings of N similar users for the unseen movie r_{vi}, and their average ratings r_v, the average ratings given by the active user r_u. These predicted value for users show better results as they are personalised for the user. Hence we plan to incorporate the location of the users to our model to check for improvements in the predicted ratings.

$$\text{sim}(u,v) = \frac{\sum_{i \in I}(r_{ui} - \overline{r_u})(r_{vi} - \overline{r_v})}{\sqrt{\sum_{i \in I}(r_{ui} - \overline{r_u})^2}\sqrt{\sum_{i \in I}(r_{vi} - \overline{r_v})^2}}$$

Figure 2: Pearson Correlation

$$\text{pred}(u,i) = \overline{r_u} + \frac{\sum_{v \in N} \text{sim}(u,v) \cdot (r_{vi} - \overline{r_v})}{\sum_{v \in N} \text{sim}(u,v)}$$

Figure 3: Making Prediction

The paper "Hybrid Movie Recommender System based on Resource Allocation" talks about solving the cold-start problem for

new movies so as to improve the recommendations by implementing a hybrid movie recommender system based on resource allocation. Users are grouped into different clusters based on their age and gender and a recommender model is developed which is a combination of content based methods and collaborative filtering methods. Content based methods allow us to recommend based on the user browsing history and preferences whereas collaborative filtering allows us to implement the recommendations based on other users in that cluster. The implemented hybrid recommendation model works in two phases. The offline phase groups the users into various clusters of different age group and gender. With the MovieLens data the model groups the users into four clusters of male and female users with the age limit of 20 to 39 and the age limit of 40-60. This phase maps the similar users based on movie ratings using the self-organizing map clustering method. The online phase then uses a three layer Artificial Neural Network to identify the cluster the active user belongs to. The similarity between the user and the users in the cluster is calculated with the Pearson similarity function (2). Then using a Content-based method it detects the preferred genre of that user and then using Collaborative Filtering it predicts the ratings for the unrated movies. For evaluating the hybrid model they use the Mean Absolute error (Figure 6) matrix on the five-fold cross validation algorithm to get the average error over five iterations. It was observed that by utilizing contextual information for predicting the ratings we could solve the cold-start problem for new movies. Using the same parameters we can also look at how the predicted ratings improve.

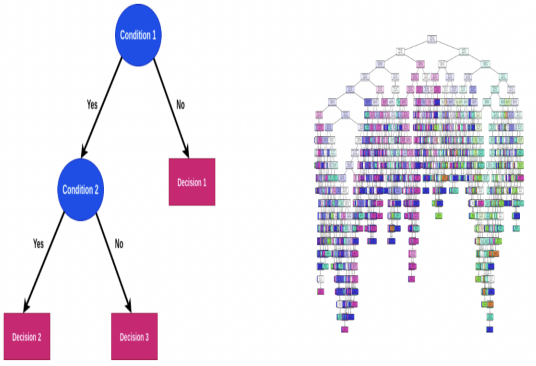
The paper "A Random Forest Approach for Rating based Recommender System" uses K-means++ algorithm and Random Forest for predicting recommendations based on user Ratings. K-means++ is an unsupervised learning algorithm which is used to cluster the data. A cluster centroid is chosen at random from the data and its distance is calculated from all other data points. The point at the farthest distance is chosen as the centroid for the next cluster and the process is repeated for k clusters. K-means++ algorithm performs better by selecting optimal clusters as compared to k-means algorithm. They have used the MovieLens data and formed k clusters based on user ratings. After clustering the ratings for unseen movies are predicted using Random Forest Classifier. The predictions are evaluated by mean squared error matrix. We try to compare the results we get on Random Forest Classifier when we include more features such as user's age, gender, demographics and movie details for training the model.

4 METHODOLOGY

Since the paper by Andrej Kosir experimented with the effect of use of the important contextual variables through collaborative filtering, we wanted to test whether the same results hold for other recommendation systems. Hence, we developed two models, a decision tree model and a random forest model. Decision trees are a machine learning algorithm that can be used for classification as well as regression. Decisions are made on conditions on the features. When mapping out a decision tree, the internal nodes of the tree represent the conditions on the features and the leaf nodes represent the decisions made based on the conditions on those features. In other words, 'a decision tree is a graphical representation of all possible solutions to a decision based on certain conditions' [**?**]. Decision Trees follow the CART (Classification and Regression Trees) algorithm. This algorithm rules that when selecting conditions on the features at any step of the decision tree, we aim to separate the labels to the highest purity. This purity can be measured by the Gini index. The Gini index is a metric to measure impurity and can be calculated by the formula: Gini Impurity = $1 - P^2_{Class1} - P^2_{Class2}$. This impurity is calculated for each decision made on the classes or labels, after which the weighted average of the Gini impurities is calculated. This acts as the loss function for the machine learning algorithm, where it seeks to structure the decision tree in such a way that the Gini index is minimized. For a completely pure distribution, the Gini impurity is zero.For our decision tree models, we relied on the Gini index for selecting variables and conditions to make decisions. To select values to make conditions, the following steps are performed:

- sort the dataset based on the numerical valued feature from low to high
- calculate the average of the adjacent pairs of numeric values
- pick average values as cutoff for the condition that minimizes the Gini Impurity index.

For a non-numerical feature, every value is tried and fitted to check which provides the minimum impurity index and that value is used to set a condition on the categorical feature. We modeled a decision tree for each of our four subsets of data. The training data for each subset was used to train their respective trees and the testing data subsets were used for testing accordingly. For each tree, we calculated the mean absolute error and mean squared error for further analysis.



(a) Example of a decision tree's conditioning

(b) Decision tree model generated

Figure 4: Decision Tree

For the next portion of our project, we ran the four subsets of data on a Random Forest Model. Random Forest Regression (RFR) is a supervised learning algorithm that combines predictions from a specified number of decision trees to make more accurate predictions than a single decision tree model. In the case of numerical data, RFR makes predictions based on the mean of individual predictions, and in the case of categorical data the *mode* of individual predictions. Theoretically, the RFR model should provide more accurate

predictions than the Decision Tree model, and this is based on the Strong Law of Large numbers which states that the average of the results obtained from a large number of trials tends to the true expected value as the number of trials gets larger. So in our case, the predictions made by RFR should get closer to the true value as we increase the number of estimators (decision trees) in our model. This is because decision trees have random sampling, and when we take a combination of decision trees in a random forest, the predicted value stabilizes and gets closer to the true value. We used the same testing and training split as in the Decision Tree model, and test the models on S1, S2, S3, and S4. The results are summarized in Table 2 of the *Results* section.

5 EXPERIMENTAL RESULTS

For the evaluation of our models, we will use metrics such as mean-squared error (Figure 5) and mean absolute error (Figure 6) as these are the most commonly used evaluation metrics for recommender systems.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Figure 5: Mean Squared Error

We calculate the mean squared error loss for all four subsets with different parameters as mentions in the table (Figure 4).

Type	MSE of Tree	MSE Random Forest
No Additional User Data	2.517	1.197
All User Data	1.783	0.886
Important Contextual Data	1.520	0.888
Important Contextual and Inconclusive Data	1.615	0.872

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Figure 6: Mean Absolute Error

Further we also test the data with the mean absolute error loss for all four subsets with different parameters. We notice that with no additional personalized information of user data we get the highest error. With all parameters in the Comodo dataset we see a drop in the error which suggest that with adding more parameters we improve the recommendation predictions for movies. Now if we add the contextual information to the model like emotions, mood we see a further drop in error.

Type	MAE of Tree	MAE Random Forest
No Additional User Data	1.191	0.898
All User Data	0.952	0.723
Important Contextual Data	0.828	0.724
Important Contextual and Inconclusive Data	0.872	0.720

6 CONCLUSION

From our observations of the error outputs of our recommendation models, we can clearly see that by incorporating additional contextual information decreases errors for both decision tress and random forests, implying that our model performance improves by this addition. This performance is further improved by selectively using only important contextual information, while using inconclusive variables improved errors slightly for random forests but did not do the same for decision trees. Theoretically, it can be said that random forests perform more accurately than decision trees, therefore the inconclusive data could likely be improving the movie recommendation performance. However, further research is required to support the theory that the inconclusive variables help with recommendation. Furthermore, since this was a small dataset, running our models on a larger dataset when collected will give us more insight into the performance change by adding contextual information in the future.

To summarize, our research question was to test whether incorporating additional contextual information about users (apart from the variables used by MovieLens dataset)improves the performance of our movie recommendation models. We created four subsets on data by sampling contextual variables that were deemed 'important', 'inconclusive' and 'irrelevant' by the curators of the LDOS Comoda dataset. The characteristics of these variables were decided after statistical analysis and the outcomes of using these variables on collaborative filtering movie recommendation models supported the aforementioned characteristics. We wanted to check if these results would corroborate on other models and hence chose to run these datasets on alternate machine learning algorithms: decision trees and random forests. We picked these models due to the sparse and small nature of the dataset, which made it incompatible for algorithms like neural networks. We evaluated our models based on the metrics mean absolute error and mean squared error for its effectiveness and usability. After observing the results, we concluded that incorporating the additional contextual information does infact improve the performance of our movie recommendation models, especially for the 'important' contextual variables. This insight could have practical applications in song recommendation systems and online shopping algorithms. We understand the challenge of curating such information on a large scale and look forward to exploring it in the future.

REFERENCES

1. Kasra Madadipouya, Department of Computing and Science, Asia Pacific University of Technology Innovation, A Location Based Movie recommender System using Collaborative Filtering,

https://arxiv.org/pdf/1508.01696.pdf

2. Mostafa Khalaji, Chitra Dadkhah, Joobin Gharibshah, Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran, University of California – Riverside, CA, Hybrid Movie Recommender System based on Resource Allocation, https://arxiv.org/pdf/2105.11678.pdf

3. Konstantinos Bougiatiotis,Theodoros Giannakopoulos, Institute of Informatics and Telecommunications, National Center of Scientific Research Demokritos, Greece, Multimodal Content Representation and Similarity Ranking of Movies, https://arxiv.org/pdf/1702.04815.pdf

4. Gediminas Adomavicius, Bamshad Mobasher, Context-aware recommender sys- tems, AI Magazine, (2011), https://ojs.aaai.org//index.php/aimagazine/article/view/2364

5. A. Livne, E. S. Tov, A. Solomon, A. Elyasaf, B. Shapira, and L. Rokach, Evolving context- aware recommender systems with users in mind, CoRR, abs/2007.15409 (2020), https://arxiv.org/abs/2007.15409.

6. A.Odic ,M.Tkalc ˇ ic ˇ ,J.F.Tasic ˇ ,and A.Ko ˇ sir,Predicting and Detecting the Relevant Contextual Information in a Movie Recommender System, Interacting with Computers, 25 (2013), pp. 74–90, https://doi.org/10.1093/iwc/iws003, https://doi.org/10.1093/iwc/iws003

7. A.Roman, Unsupervised classification project: Building a movie recommender with clustering analysis and k-means, (2019), https://towardsdatascience.com/unsupervised-classification-project-building-a-movie-recommender-with-clustering-analysis-and-4bab0738efe6.

8. C. Saluja, Collaborative filtering based recommendation systems exemplified, (2018),https://towardsdatascience.com/collaborative-filtering-based-recommendation-systems-exemplified-ecbffe1c20b1.