A Novel Approach for solving New User Problem in Recommender Systems

Kapil Agarwal* Mohit Bakshi* Durga Toshniwal

Department of Computer Science & Engineering
IIT Roorkee
Roorkee, India 247667
{kapil6442, mohitbakshi2205, durgatoshniwal}@gmail.com

ABSTRACT
A good recommendation system should be able to generate meaningful recommendations to its users for the items and products that are of their interest. Existing techniques based on collaborative filtering and content-based filtering do not tackle the problems posed by new users to a recommendation system, since without previous preferences of a user, it is not possible to find users with similar interests or construct a content-based profile of the user. In this paper, we propose a novel approach to solve the new user problem with a better success rate than giving random recommendations. We consider a case study on movie dataset and develop a hybrid model that uses both user attributes and product features to recommend items to the user. The model is based on the assumption that similar users give similar ratings to the products and recommends items based on the predicted rating of a product by the new user.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database – *Data mining.*

General Terms
Design, Experimentation, Human Factors, Theory.

Keywords
Data Mining, Clustering, Classification, Recommender Systems.

1. INTRODUCTION
Data Analytics has opened a wide range of opportunities in various fields. Applications making use of insights and trends on social media are now becoming more and more popular. The scale at which data is being generated is beyond what a single application can comprehend and make use. With over 50 million tweets per day [17] and more than a billion people using their Facebook account once a month [16] the possibilities with mass monitoring applications are limitless.

Business Organizations are taking a particular interest in leveraging these new public data channels. By finding out the needs and complaints of their customers, these organizations can obtain useful product information and market potential to align their strategies accordingly. Combining social media data with predictive models, businesses can predict financial metrics and work out the return on investment for the social media campaigns. Besides marketing and finance, customer relationship management is also being benefited by using social data as the companies can store information corresponding to individual customers and use it to handle customer relationships.

A particular industry, whose profit/loss is widely dependent on Word Of Mouth and public mood, is the movie industry [10]. There is an increasing number of users who are looking for movie reviews online because it is both fast and varied. A recommendation to watch a movie by an influential member of the community might have widespread positive effect for movie sales (this is particularly true for twitter users having thousands and millions of followers) while a negative sentiment may carry repercussions for the movie reputation and sales. Thus it becomes increasingly important to analyze how data gathered from various social media channels can be leveraged to find if a relationship between movie sales and popularity exists with the sentiment displayed by public on these channels.

The success of a movie can be highly attributed to the demographics of its viewers as well. Different people may perceive a movie in different ways. For example, a movie portraying the Indian Freedom Struggle may be very popular among people in the Indian sub-continent but may not be so in other regions of the world. Similarly a movie about computers may be more popular among college students and programmers than an artist or politician. The popularity of a movie can also depend on the geographical distribution of the viewers. An economically backward area may lack sufficient number of theatres or people may not be able to afford paying for watching a movie. Such characteristics of viewers can affect the success of a movie and its popularity amongst the target audience.

Data can exist in various forms. Data can be structured (SQL), unstructured (text, videos, audio files) or semi-structured (XML, JSON). Data can be qualitative / nominal (favorite color, place of birth, type of car) or quantitative. Quantitative data can further be discrete / ordinal (shoe size, number of siblings) or continuous / numeric (height, mass, length). Data can also classified as only number data or text data or a combination of both. Such heterogeneous raw data, which cannot be classifies into a single category of type of data is referred to as complex data. In this paper, we use a *complex* data set which has both numeric and nominal data, as well as contains both number and text data.

We have used several data mining techniques like classification, clustering, association rule mining, and statistical analysis techniques to develop a mathematical model that best fits the given dataset. During this analysis, we aim to establish a relationship between product popularity and data obtained from the users like user ratings of the movies and characteristics of the user. The popularity of a product can be measured in various ways like sales, revenue, profit or even buzz on social media, sentiment of the reviews of the product, etc. We however use the ratings of the product provided by the user as a measure of its popularity. The model predicts the rating of the product by a user so as to recommend that product to the user.

The predictive model leverages such data and predicts the product popularity. The dataset that we use is obtained from GroupLens, a social computing research group at University of Minnesota which has a subproject named "MovieLens", under which they collect free, non-commercial data from individual users and provide personal recommendations.

The paper is organized as follows. Section 1 introduces the topic of our project. Section 2 presents the study of various related work that has already been done to solve similar problems on the MovieLens dataset. Various algorithms and toolkits that have been used to develop the model are also reviewed with regard to their features and availability. Section 3 describes the data set we have used for implementing and testing the proposed model. Section 4 includes the details of our approach to solving the problem, the models and heuristics we have developed and their implications. In section 5, we discuss the results of our analysis. Finally, section 6 discusses what we conclude and scope of future work.

2. RELATED WORK

Evaluation of product popularity based on user ratings has been given considerable importance recently, with researchers employing a wide variety of techniques and heuristics to model the situation. In this section we present some of the literature on this subject that we have come across.

2.1 Collaborative Filtering

Jung [8] proposes an attribute reduction-based mining method to efficiently select the long-tail user groups i.e. users that have shown preferential patterns concentrated to a small set of attributes. The paper assumes that a group of users in long tail should be regarded as the professional experts in corresponding domains, and employed their opinions to provide recommendations to users in short head. Short Head user Groups show inter-mixed preferential patterns. In the paper, ''dominant coverage'' score of each attribute from the user ratings is measured for efficiently justifying which attribute is strongly related to the user preference. A dominant attribute of a user is selected when its dominant coverage is significantly larger than the others. After computing the dominant attributes of each attribute from the users, they find the long tail user groups and reuse the user ratings given by LTuG (Long Tail user Group) for giving recommendations.

Marlin [12] proposes the User Rating Profile model (URP) which is a generative latent variable model for rating-based collaborative filtering. In the URP model, each user is represented as a mixture of user attitudes, and the mixing proportions are distributed according to a Dirichlet random variable. For generating the rating for each item, a user attitude for the item is selected, and then a rating according to the preference pattern associated with that attitude is selected. The procedure they have used for parameter estimation is a variational expectation maximization algorithm based on free energy maximization.

2.2 Content-based Filtering

Kim & Kim [9] present a model-based recommendation algorithm that uses multi-level association rules to solve the problems of data sparseness and scalability in collaborative filtering methods. In this method, the authors first find the category c_i to which the preferred item belongs. If there is a category association rule of the form $c_i \Rightarrow c_j$, then they give certain amount of preference to all items that belong to category c_j.

Active learning strategies identify the most informative set of training examples through minimum interactions with the users. Harpale & Yang [3] propose an extended form of Bayesian active learning and use the Aspect Model [4, 5] for collaborative filtering. They personalize active learning for the user by querying for only those items which the user can provide rating for. The goal of active learning in CF is to obtain ratings for more items from the new user. Instead of randomly selecting movies for rating from the user, active learning algorithms minimize the number of such queries required to learn a stronger user-model.

3. DATASET

GroupLens is a research lab in the Department of Computer Science and Engineering at the University of Minnesota, Twin Cities specializing in recommender systems, online communities, mobile and ubiquitous technologies, digital libraries, and local geographic information systems. MovieLens (http://movielens.org) is a web site that helps people find movies to watch. It has hundreds of thousands of registered users. GroupLens Research has collected and made available rating data sets from the MovieLens web site (http://movielens.org). We have used the "MovieLens 1M" dataset [18] for this project. It contains 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000. There are mainly three files in the dataset, namely, ratings file, users file and movies file. The entity relationship diagram of the dataset is described in Figure 1.

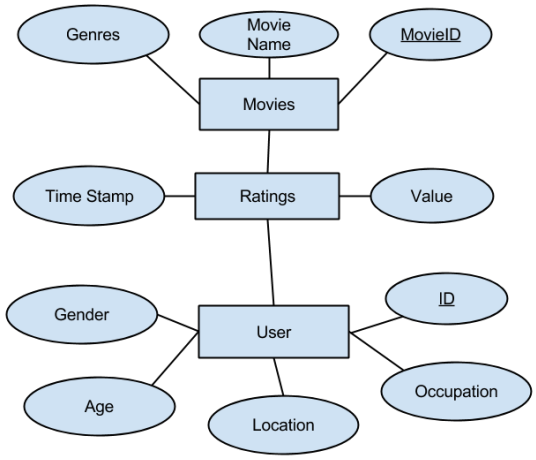

Figure 1. Entity relationship diagram of the dataset

In the ratings file, there is a tuple for each user u who has rating a movie m with a rating r at time t. Ratings are made on a 5 whole star ratings scale. Also, each user has at least 20 ratings.

In the users file, demographic information of the users like gender, age, occupation and location are given. All these user features are categorical attributes.

The movies file contains the genres for each movie that has been rated by the users. A movie can have more than one genre and vice versa.

4. PROPOSED WORK

Figure 2 shows the workflow that we have proposed and followed to build a model that solves the problem at hand.

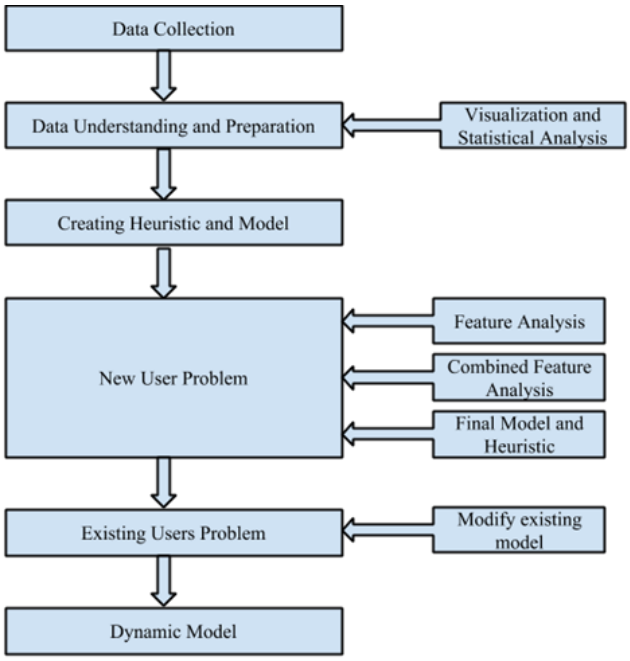

Figure 2. Proposed Workflow

There are various measures to represent the popularity of a product. Some monetary measures are sales, price, profit, revenue, etc. of a product or brand. However, there are some non-monetary measures as well such as data available on social media like Facebook, Twitter, YouTube, IMDb, etc. Our case study focuses on movies as the product for which we aim to develop a model which can predict the popularity of a movie among a set of viewers.

The next step is to familiarize ourselves with the data, identify any problems in the data quality, and discover insights into the data that aid to form hypotheses about any hidden information. The preliminary analysis includes finding the distribution of the user attributes across all users and the distribution of movie attributes across all the movies, which we refer as visual and statistical analysis of the data. Plotting the locations of all the users on a map provided a geographical view of the dataset and helped assess the usability of each attribute in solving the problem. After the preliminary analysis, initial raw data is constructed into the final data set that will be fed into the modeling tool.

The MovieLens dataset contains only the ratings of various movies by the users. There is no direct relationship between the attributes of the users and the attributes of the movies. To solve this, we developed three heuristics that help in establishing an output value, a movie attribute, for each tuple consisting of user attributes. After experimenting with several classification and clustering algorithms and calibrating the parameters to obtain optimum values, we develop a model that suits best for the given dataset. Our model for determining the popularity or rating of a movie by a user is tested based on these heuristics.

4.1 User-Genre Preference

The movie feature that we relate to the user demographics is movie genre. With a list of 18 genres given in the dataset, we need to find a heuristic that assigns to each genre and user a number/score that measures that user's preference for that genre. For example, User-Genre Score (UGS) of a user 'Alice' to genre 'Comedy' is 0.4 and to genre 'Drama' is 0.6 while that of user 'Bob' for the genre 'Comedy' is 0.7 and for genre 'Drama' is 0.2. This should suggest that Alice prefers Drama over Comedy and Bob prefers Comedy over Drama. Bob prefers Comedy more than Alice prefers Drama. The UGS may or may not be normalized over all genres for a given user. We develop three heuristics for calculating the UGS as described in the following sections.

4.1.1 Rating Independent

Let the user i have a user-genre score for a genre g as follows:

$$UGS_{\mathrm{i}}^{\mathrm{g}} = \frac{1}{N} X \sum_{m=1}^{N_g} \frac{1}{|G_m|};\ \forall\, m \mid g \in G_m$$

where $m = 1$ to N_g is the iterator over all movies user i has seen. G_m is the genre list for movie m. N is the total number of movies user has seen. Henceforth, we refer to this as 'Heuristic 1'.

This heuristic makes certain assumptions described as follows. Initially, UGS for each genre g is 0 and each movie m user i has seen is given a score of 1. Then we iterate over all movies user i has seen and traverse over the genre list G of each movie. For each genre g in G, we add a score of $1/|G|$ to the score corresponding to genre g. This heuristic assumes that if a movie has been tagged with n number of genres, then each genre has contributed $1/n$ to the score of that movie. The reason why we have chosen $1/|G|$ instead of considering rating r and assigning each genre a score of $r/|G|$ is following - consider a user who likes genre '*drama*', we can safely assume that this user has seen a lot of drama movies. So this user is going to be critical in his or her analysis of any new drama movie. So it is not necessary that the user gives a higher rating to a movie just because it has the genre the user likes. Therefore using the rating of the movie to test if the user likes its genre must not be considered a convention.

4.1.2 Rating Dependent Genre Independent

Let the user i have a user-genre score for a genre g as follows:

$$UGS_{\mathrm{i}}^{\mathrm{g}} = \left. \sum_{m=1}^{N_g} \frac{r_m}{|G_m|} \right/ \sum_{m=1}^{N_g} r_m \ ;\ \forall\, m \mid g \in G_m$$

where r is the rating given by the user to movie m and g is genre in the list of movie m. N is the total number of movies the user has seen and N_g is the number of movies that user has seen that contains genre g. Henceforth, we refer to this as 'Heuristic 2'.

This heuristic assumes that the rating given by the user to a movie affects his or her preference for all genres in that movie. For example, if a user is giving a rating of 4 to a movie m_1 having 2 genres and a rating of 2 to a movie m_2 having 2 genres, then movie m_1 genres get twice the score as compared to movie m_2 genres. Also if movie m_1 has 4 genres instead of 2, then the score contributed by each genre is only $1/4$ of the total rating of movie, so that the each genre of m_1 gets the same score as genres of m_2.

4.1.3 Rating Dependent Genre Dependent

There are several limitations with the above heuristics. Heuristic 1 does not take into account the rating of the user. Since we need to predict what rating a user might give to a movie which he/she has not yet seen, therefore this heuristic which assigns a value of 1 to

each movie and distributes it among all its genres is not very useful for predicting the rating.

The problem with heuristic 2 is more subtle and implicit. Heuristic 2 divides the rating given to a movie among its genres equally and then finds the sum of all such contributions for each genre. Thus the final UGS for a genre is not only dependent on the average rating given by the user to all movies containing that genre but also dependent on the proportion of movies seen by the user that contains that genre. Thus if a user has not seen many movies having 'comedy' as compared to 'drama' even though the average rating given to comedy is more than that of drama, then this heuristic will penalize comedy. So when a new comedy movie and another drama movie arrives, this heuristic will predict a higher rating for drama movie as compared to comedy movie even though we know that on an average the user gives more rating to comedy movies.

Also, we are assigning a single genre label to each of the user even though the user may very closely prefer another genre. Thus we need to find a heuristic that can distribute the preference of a user over all genres while taking into the account the limitations of the above 2 heuristics. To counter above limitations, we propose another heuristic and combine it with a rating predictive model which is described below.

Let the user i have a user-genre score for a genre g as follows:

$$UGS_{\mathrm{i}}^{\mathrm{g}} = \sum_{m=1}^{N_g} \frac{r}{N_g};\ \forall\, m \mid g \in G_m$$

where r is the rating given to movie m and N_g is the number of movies in which genre g is present. Thus m iterates over all those movies which contain genre g. Henceforth, we refer to this as 'Heuristic 3'.

This heuristic finds the average rating given by a user to all the movies in which the particular genre is present. The difference from heuristic 2 is that here instead of normalizing the rating for a genre over all the movies user has seen, we normalize it over only those movies which contain that particular genre. So this heuristic for measuring preference ignores the volume of movies seen by the user. For example, user i has seen 4 movies having '*comedy*' as a genre and the average rating over these 4 movies is 3.6 while the user i has seen only 1 movie having genre 'drama' as a genre and the rating given to this movie is 4.5 then this heuristic assigns higher score to drama than comedy, ignoring the fact that the user i has seen more number of comedies than drama.

4.2 Predictive Model

Now that we have formalized the preference of a user over genres, we need to formulate how we can use these heuristics to predict the rating a user will give to a movie he or she has not yet seen and also how we can predict the behavior of a user whose history of ratings is not available but some user demographics are given.

Firstly, the UGS of each user for each genre can be represented in a 2D matrix as shown in Table 1. To maintain dynamicity in the model, we need to store another matrix that counts the number of movies associated with each genre that the user has seen as shown in Table 2.

As the user keeps watching and rating more movies, both UGSM and GCM can be updated as follows. For heuristic 3, for each genre j in the movie m that user i has just given a rating r:

Table 1. UGS Matrix (UGSM)

Users	Genre 1	Genre 2	Genre 3	...	Genre m
U_1	UGS_1^1	UGS_1^2	UGS_1^3	...	UGS_1^m
U_2	UGS_2^1	UGS_2^2	UGS_2^3	...	UGS_2^m
…	…	…	…	...	…
U_n	UGS_n^1	UGS_n^2	UGS_n^3	...	UGS_n^m

Table 2. Genre Count Matrix (GCM)

Users	Genre 1	Genre 2	Genre 3	...	Genre m
U_1	N_1^1	N_1^2	N_1^3	...	N_1^m
U_2	N_2^1	N_2^2	N_2^3	...	N_2^m
…	…	…	…	...	…
U_n	N_n^1	N_n^2	N_n^3	...	N_n^m

$$UGS_{\mathrm{i}}^{\mathrm{j}} = \left(UGS_{\mathrm{i}}^{\mathrm{j}}\ X\ N_{\mathrm{i}}^{\mathrm{j}} + r \right) \Big/ \left(N_{\mathrm{i}}^{\mathrm{j}} + 1 \right)$$

$$N_{\mathrm{i}}^{\mathrm{j}} = N_{\mathrm{i}}^{\mathrm{j}} + 1$$

In accordance with heuristic 3, the UGS score of a user u for genre g represents the average rating that the user has given to movies having genre g in its genre list. We can intuitively assume that for any new movie the user maintains his or her previous behaviour.

So for a user U_x and a movie having genres g_j, g_l, g_m and so on, we can say that the rating given to movie m in accordance with the information that the movie has:

- genre g_j is $UGS_{\mathrm{x}}^{\mathrm{j}}$
- genre g_l is $UGS_{\mathrm{x}}^{\mathrm{l}}$
- genre g_m is $UGS_{\mathrm{x}}^{\mathrm{m}}$

We can assume that each of this information is equally important and hence the predicted rating of the movie will be the average of

all ratings predicted by this information. Therefore, predicted rating according to heuristic 3 is given by:

$$r_{predicted} = \sum_{g=1}^{|G|} UGS_{\mathrm{i}}^{\mathrm{g}} \Big/ |G|$$

As shown in figure 3, function $f()$ can be any aggregator. Additive, averaging, multiplicative functions are some examples. As described above, for heuristic 3, using averaging function makes sense.

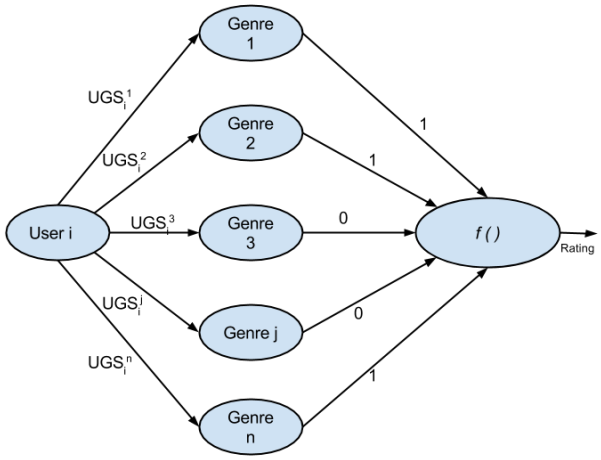

Figure 3. Movie m having genres 1, 2 and n (edges having 1)

4.3 Cold-start Problem

The problem addressed in this paper can be divided into two parts. The first part is the new user problem for which we develop a novel model. The existing models like MovieLens and IMDb require users to input ratings of at least 15 movies before movies can be recommended by the user. However, we attempt to give movie recommendations to users based only on their characteristics which they input only once while registering for the service. The cold start problem can be stated as: given a new user whose history of rating is not available, predict his or her rating for a given movie.

In accordance with the model, we need to establish UGS for this new user for each of the genre. Once we have this UGS vector, we can predict the rating of a movie as explained earlier. We assume here that we are given user demographics i.e. certain basic information about the user such as gender, occupation, age, location etc. Using this information we can find users who are similar and we can utilize their UGS vectors to find the possible value of UGS for this user.

Given attributes of a user (demographics such as age, gender, occupation etc.) we need to find a set of users in the dataset who are similar to this user. Here we use the idea of unsupervised learning i.e. clustering the user base on user features only without labelled output genre preferences. By clustering the user base on demographic information we can find a set of users who are similar to each other and when a new user arrives, we can find the closest cluster and assign this new user to that cluster.

Table 3. User-Attribute Matrix

User	Attribute 1	Attribute 2	…	Attribute m
U_i	$A_1(i)$	$A_2(i)$	…	$A_m(i)$
U_j	$A_1(j)$	$A_2(j)$	…	$A_m(j)$

Given a user U, whose attributes A_1, $A_2 \ldots A_n$ are given, as shown in table 3, we have to find the set of users who are similar to this user in accordance with these attributes. To find the distance between users, we consider the user dataset. The attributes available to us are all categorical in nature. There are many clustering algorithms that exist for performing clustering on numeric data like K-means [11], EM algorithm, Hierarchical clustering, etc. However, we need to perform clustering on categorical data which has a different structure than the continuous data. The distance functions in the continuous data might not be applicable to categorical data. So, algorithms for clustering continuous data cannot be applied directly to categorical data. Some clustering algorithms for categorical data are K-modes [6, 7] (which is a modification of the K-means algorithm), AutoClass (which is based on EM algorithm), ROCK [1] and CLOPE [15]. The most common distance metric used for categorical data is the overlap distance. For users U_i and U_j having categorical attributes, we define:

$$X_k(i,j) = \begin{cases} 1\,; if\ A_k(i) \neq A_k(j) \\ 0\,; if\ A_k(i) = A_k(j) \end{cases}$$

$$Distance\ (i,j) = \sum_{k=1}^{m} X_k(i,j)$$

The overlap distance is similar to hamming distance between the categorical attribute vectors. Using this distance metric we use K-Modes Clustering algorithm to find sets of similar users. In K-modes, we choose k cluster modes and iterate until convergence. However, it has a fundamental flaw that the partition is sensitive to the input order i.e. the clustering results would be different for the same data set if the input order is different. K-Function $f()$ can be any aggregator. Additive, averaging, multiplicative functions are some examples. For heuristic 3, using averaging function makes sense. K-Modes clustering is necessary because the attributes are all categorical in nature.

When we obtain a new user, we find the cluster whose mode is the closest to this user and find the average UGS for the entire cluster. We assume that this average UGS vector defines the preference of this new user towards each genre. We use this UGS vector to predict rating for this user for each movie in the movie database and recommend movies that have the highest predicted rating.

5. RESULTS AND DISCUSSIONS

5.1 Feature Analysis

On obtaining the UGS for each user and genre, we perform individual feature analysis for each feature i.e. gender, age and occupation. Feature analysis consists of analyzing the relationship between each user attribute and the corresponding movie attribute obtained from the heuristics we have developed. We obtain the following entropies for each user attribute:

$$Entropy(Gender) = 0.8594$$

$$Entropy(Age) = 2.4772$$

$$Entropy(Occupation) = 3.9971$$

We then find the information gain for each attribute of the user for heuristics 1 and 2. We find that the information gains are insignificant as compared to the original entropy of the attributes, so using individual features to predict genre or vice-versa is not feasible. Therefore, we move over to combined-feature analysis.

Decision tree learning can build classification and regression models in the form of a tree structure. While a dataset is broken down into smaller and smaller subsets, an associated decision tree is incrementally developed at the same time. The result obtained after the model has learnt is a tree with decision nodes and leaf nodes. A decision node has two or more branches. The leaf node represents a classification or decision. Decision trees can handle both categorical and numerical data. A decision tree can be transformed to a set of rules simply by mapping from the root node to the leaf nodes one by one.

ID3 [13] by J. R. Quinlan is the core algorithm for building decision trees. It employs a top-down, greedy search through the space of possible branches with no backtracking. Entropy and Information Gain are the metrics used to construct a decision tree in ID3. A decision tree is built top-down from a root node. The data is partitioned into subsets that contain instances with similar values (homogenous). ID3 algorithm uses entropy to calculate the homogeneity of a sample.

While there are several motivations for using Information Gain as a quality measure, there are also several limitations. One of the problems is that it is biased towards choosing attributes with a large number of values. This may result in over-fitting i.e. selection of an attribute that is non-optimal for prediction. One trick to avoid this specific problem is to use Gain Ratio [14]. A small modification of the information gain gives us the gain ratio that reduces its bias on high-branch attributes. When data is evenly spread, gain ratio should be large and small when all the data belongs to one branch. Gain ratio takes into account the number and size of branches while choosing an attribute. It corrects the information gain by taking the intrinsic information of a split into account. As intrinsic information gets larger, the importance of an attribute decreases. However, there are problems with gain ratio as well like, it may overcompensate i.e. it may choose an attribute just because its intrinsic information is very low. The fix for this would be to consider only the attributes with greater than average information gain first and then compare them on gain ratio.

For performing the preliminary analysis as well as the feature analysis, we have used WEKA [2] toolkit which is a collection of machine learning algorithms that can be directly applied to the dataset.

5.1.1 Heuristic 1

After applying heuristic 1 and obtaining the corresponding user-genre preferences, we obtain the information gain and gain ratios of each of the user attributes.

$$Entropy(Genre) = 2.2192$$

Table 4. Ratios for heuristic 1

Attribute	Information Gain	Gain Ratio
Gender	0.0369	0.0429
Age	0.0551	0.0223
Occupation	0.0791	0.0198

Table 5. Classification results for heuristic 1

Technique	Correctly Classified Instances (%)	Root Mean absolute error
Simple Naïve Bayes	45.5132	0.1948
Naïve Bayes	45.5132	0.1948
Neural Network	43.4106	0.1972
ID3	42.9967	0.198
J48	44.4868	0.1957

5.1.2 Heuristic 2

After applying heuristic 2 and obtaining the corresponding user-genre preferences, we obtain the information gain and gain ratios of each of the user attributes.

$$Entropy(Genre) = 2.1998$$

Table 6. Ratios for heuristic 2

Attribute	Information Gain	Gain Ratio
Gender	0.0369	0.043
Age	0.0538	0.0217
Occupation	0.0811	0.0203

Table 7. Classification results for heuristic 2

Technique	Correctly Classified Instances (%)	Root Mean absolute error
Simple Naïve Bayes	46.4238	0.1938
Naïve Bayes	46.4238	0.1938
Neural Network	43.9735	0.1963
ID3	44.9834	0.1967
J48	46.1093	0.1938

As we can see from Table 4 and Table 6, the information gain and gain ratios are not of significant value, so decision tree classification may not give good results. Similarly, Naïve Bayes

classification has the basic assumption that the input attributes are independent of each other. In this dataset, we find that occupation is dependent on gender and age and so, Naïve Bayes classifier also may not give good results. The results of the various classification algorithms applied on the dataset are stated in Table 5 and Table 7.

5.2 New User Analysis

We use heuristic 3 to predict the rating of a movie by a user. We also divide the rating scale into different bins and then calculate the accuracy of the predicted model by matching the number of ratings that are correctly classified. Firstly, we divide the scale into five classes of the following ranges: 0 to 1.5, 1.5 to 2.5, 2.5 to 3.5, and 4.5 to 5. Secondly, we divide the scale into three classes of the following ranges: 0 to 2, 2 to 4, and 4 to 5. The predicted rating is then rounded off to the nearest integer and the corresponding class is assigned. We have performed 10 cross validation for all analysis and we varied the number of clusters in the unsupervised learning stage from 3 to 18. The metric that we find is the average root mean square error between the predicted rating and the actual rating.

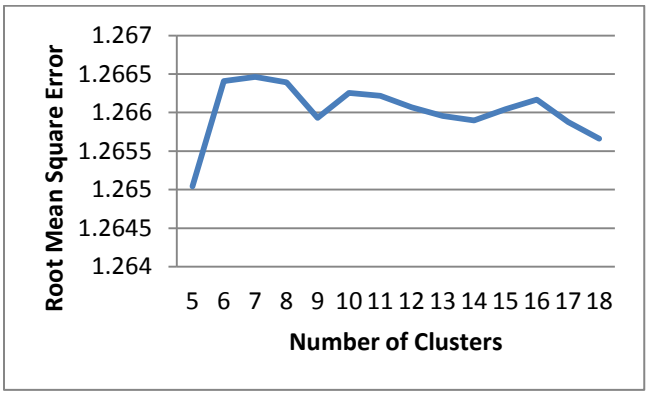

Figure 4. Five Class Validation – Root Mean Square error

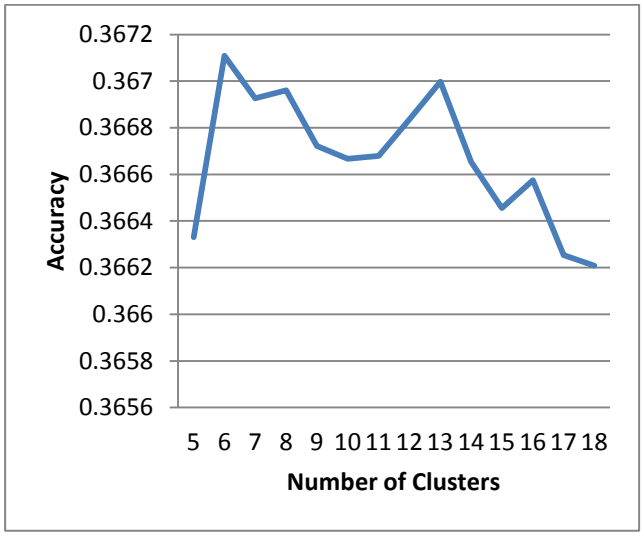

Figure 5. Five Class Validation – Accuracy

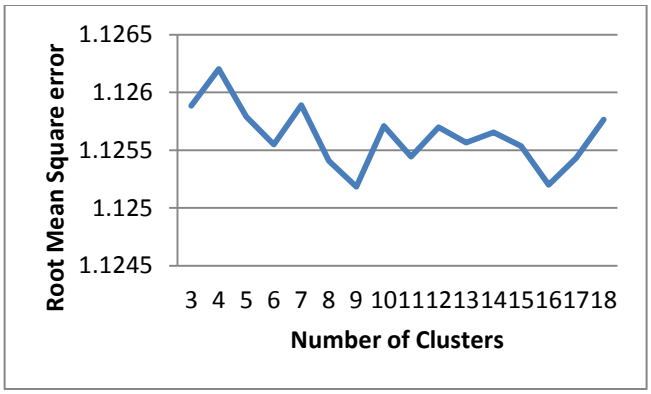

Figure 6. Three Class Validation – Root Mean Square error

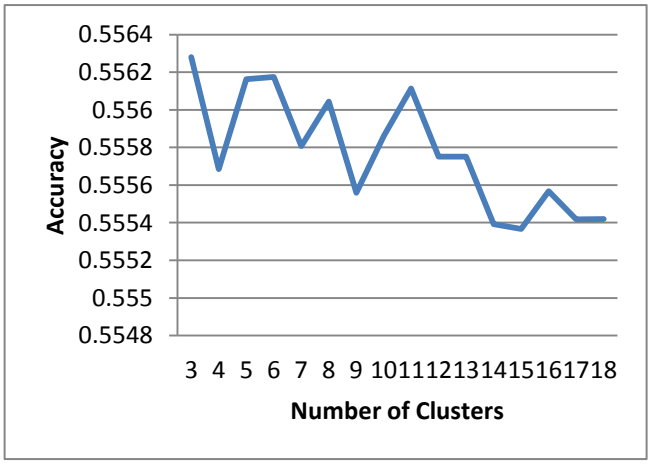

Figure 7. Three Class Validation – Accuracy

The results are much better than a random recommendation to a new user. For the three class validation, the user will be recommended a movie of his or her taste with a probability of 1/3 i.e. 33.33% in the case of random recommendation, whereas our model will recommend it with around 55% accuracy. In the case of five class validation, this probability would decrease to 20% in case of random recommendation, while our model recommends with an accuracy of around 36% which is still better. Thus we can see, our model performs better than simple random approach.

6. CONCLUSION AND FUTURE WORK

We developed three heuristics for solving the problem of predicting product popularity for a new user. We developed a novel model to find a relationship between the characteristics of a user and the ratings he or she would give to a particular product. We have done a case study on the movie dataset and obtained significant results. Heuristics 1 and 2 have some limitations which are overcome by Heuristic 3, and so the model based on this heuristic is run on the dataset to calculate the predicted ratings of the users. We performed 10 fold cross validation and obtain better results than the naïve random guessing approach.

This model can be extended to solve the existing user problem as well with slight modifications to the model to incorporate the previous ratings of products by the user. We can find association

rules between the characteristics of the product i.e. movies which enable to cluster the movies into groups. The new movie problem can also be solved using this model by finding the most similar cluster and then calculating the average rating of the movies in that cluster by a user.

Currently, the model is static in nature i.e. it works only on data currently in the system and if there is a new entry into the dataset, the model will have to be re-run to get the new results. There are several ways to overcome this. One of the ways is to re-run the model only after a certain number of new entries into the dataset assuming that a small number of new entries don't affect the accuracy of the model by a significant amount. The other method is to make the system dynamic. However, the K-Modes clustering algorithm cannot incorporate dynamicity and so other clustering algorithms will need to be explored or developed.

7. REFERENCES

[1] Guha, Sudipto, Rajeev Rastogi, and Kyuseok Shim. "ROCK: A robust clustering algorithm for categorical attributes." Data Engineering, 1999. Proceedings., 15th International Conference on. IEEE, 1999.

[2] Hall, Mark, et al. "The WEKA data mining software: an update." ACM SIGKDD explorations newsletter 11.1 (2009): 10-18.

[3] Harpale, Abhay S., and Yiming Yang. "Personalized active learning for collaborative filtering." Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2008.

[4] Hofmann, Thomas. "Collaborative filtering via gaussian probabilistic latent semantic analysis." Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. ACM, 2003.

[5] Hofmann, Thomas, and Jan Puzicha. "Latent class models for collaborative filtering." IJCAI. Vol. 99. 1999.

[6] Huang, Zhexue. "Clustering large data sets with mixed numeric and categorical values." Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining,(PAKDD). 1997.

[7] Huang, Zhexue. "Extensions to the k-means algorithm for clustering large data sets with categorical values." Data Mining and Knowledge Discovery 2.3 (1998): 283-304. APA

[8] Jung, Jason J. "Attribute selection-based recommendation framework for short-head user group: An empirical study by MovieLens and IMDB." Expert Systems with Applications 39.4 (2012): 4049-4054.

[9] Kim, Choonho, and Juntae Kim. "A recommendation algorithm using multi-level association rules." Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on. IEEE, 2003.

[10] Liu, Yong. "Word of mouth for movies: Its dynamics and impact on box office revenue." Journal of marketing 70.3 (2006): 74-89.

[11] Lloyd, Stuart. "Least squares quantization in PCM." Information Theory, IEEE Transactions on 28.2 (1982): 129-137.

[12] Marlin, Benjamin M. "Modeling User Rating Profiles For Collaborative Filtering."NIPS. 2003.

[13] Quinlan, J. Ross. "Induction of decision trees." Machine learning 1.1 (1986): 81-106.

[14] Quinlan, John Ross. C4. 5: programs for machine learning. Vol. 1. Morgan kaufmann, 1993.

[15] Yang, Yiling, Xudong Guan, and Jinyuan You. "CLOPE: a fast and effective clustering algorithm for transactional data." Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2002.

[16] http://www.statisticbrain.com/facebook-statistics/

[17] http://www.statisticbrain.com/twitter-statistics/

[18] http://grouplens.org/datasets/movielens/