Çanakkale Onsekiz Mart Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 2019:5,2, 214-232

Çanakkale Onsekiz Mart University Journal of Graduate School of Natural and Applied Sciences, 2019:5,2, 214-232

Araştırma / Research

MovieANN: A Hybrid Approach to Movie Recommender Systems Using Multi Layer Artificial Neural Networks

Sait Can Yücebaş*

Çanakkale Onsekiz Mart University, Faculty of Engineering, Department of Computer Engineering, Çanakkale, Turkey

26.07.2019 Geliş/Received, 12.12.2019 Kabul/Accepted

Abstract

The amount of data in World Wide Web is growing exponentially. Users are often lost in this vast ocean of data. In order to filter the valuable information from vast amount of data, recommendation systems are used. These systems are based on collaborative filtering, content based filtering and hybrid approaches. We combined collaborative and content-based filtering to build a hybrid movie recommendation system, MovieANN, based on neural network model. To make better recommendations in a collaborative approach, both user and movie clusters are formed. In addition to rating information, content information was also considered in the formation of the clusters. Clusters are formed according to K-Means and X-Means algorithms. Final clusters are chosen according to Davies-Bouldin Index and intra cluster distance. Homogeneity and density of the clusters are also considered. Movie and user clusters are mapped in the recommendation phase. The model is tested on a MoiveLens 1M dataset that consists of six thousand users, four thousand movies and one million ratings. Four clusters are formed to represent movie – user mappings and for each cluster, a recommendation model based on multi-layer neural network is constructed. The recommendation performance in terms of accuracy is 84.52%, 84.54% in terms of precision and 99.98% in terms of recall.

Keywords: recommendation systems, content based filtering, collaborative filtering, hybrid recommender, artificial neural network

MovieANN: Film Öneri Sistemlerine Çok Katmanlı Yapay Sinir Ağı Kullanarak Karma Bir Yaklaşım

Öz

İnternetteki veri miktarı gün geçtikçe katlanarak artmaktadır. Kullanıcılar bu geniş veri okyanusunda sıklıkla kaybolmaktadır. Bu yüksek miktardaki ham veriden önemli bilgiyi filtrelemek için öneri sistemleri kullanılır. Bu sistemler işbirlikçi filtrelemeye, içeriğe dayalı filtrelemeye ve hibrit yaklaşımlara dayanmaktadır. Bu çalışmada yapay sinir ağına dayalı hibrit bir film öneri sistemi olan MovieANN, işbirlikçi ve içerik tabanlı filtreleme

*Sorumlu Yazar (Corresponding Author): Sait Can Yücebaş
(e-posta: can@comu.edu.tr)

kullanılarak gerçekleştirilmiştir. İşbirlikçi bir yaklaşımla daha iyi öneriler yapmak için hem kullanıcı hem de film kümeleri oluşturulmuştur. Kümeler oluşturulurken rating bilgisine ek olarak içerik bilgisi de dikkate alınmıştır. Kümeleme için K-Means ve X-Means algoritmaları kullanılmıştır. Son kümeler, Davies-Bouldin Endeksi ve küme içi mesafelerine göre seçilir. Kümeler oluşturulurken homojenlik ve yoğunluk da göz önünde bulundurulmuştur. Öneri adımında film ve kullanıcı kümeleri eşleştirilir. İlgili model, altı bin kullanıcı, dört bin film ve bir milyon ratingden oluşan MoiveLens 1M veri kümesinde test edilmiştir. Film kullanıcı eşlemelerini temsil etmek için dört küme ve her küme için çok katmanlı sinir ağını temel alan bir öneri modeli oluşturulmuştur. Modelin öneri performansı doğruluk olarak % 84,52, kesinlik açısından % 84,54 ve geri çağırmada % 99,98'dir.

Anahtar Kelimeler: öneri sitemleri, içerik tabanlı filtreleme, iş birlikçi filtreleme, hibrit öneri, yapay sinir ağı

1. Introduction

Recommender systems are developed to filter vast amount of data in the web and to propose the most relevant information to the users. By this way, they save time and effort for online users (Koohi, 2017). The main idea behind the recommendation systems is to rank the information according to the user’s interest domain which is learned from the past actions of the user (Haruna et al., 2017). To achieve the goal, recommender systems are built on three main filtering approaches. These are collaborative filtering, content based filtering and hybrid approaches. Collaborative filtering focuses on finding similar users and/or products. On the other hand, content based filtering proposes a more personalized approach. Hybrid models combine these two approaches to overcome the disadvantage of both methods. Weighting, switching, mixed, feature augmentation (Attarde, 2017; Lekakos, 2008; Burke, 2002) are often used for combination of different filtering mechanism in a hybrid model.

Each main recommendation approach can be divided into subgroups, each differs with the recommendation algorithm, similarity calculation and size reduction methodologies. The Figure 1.1 represents the main and sub methods used in the recommendation systems.

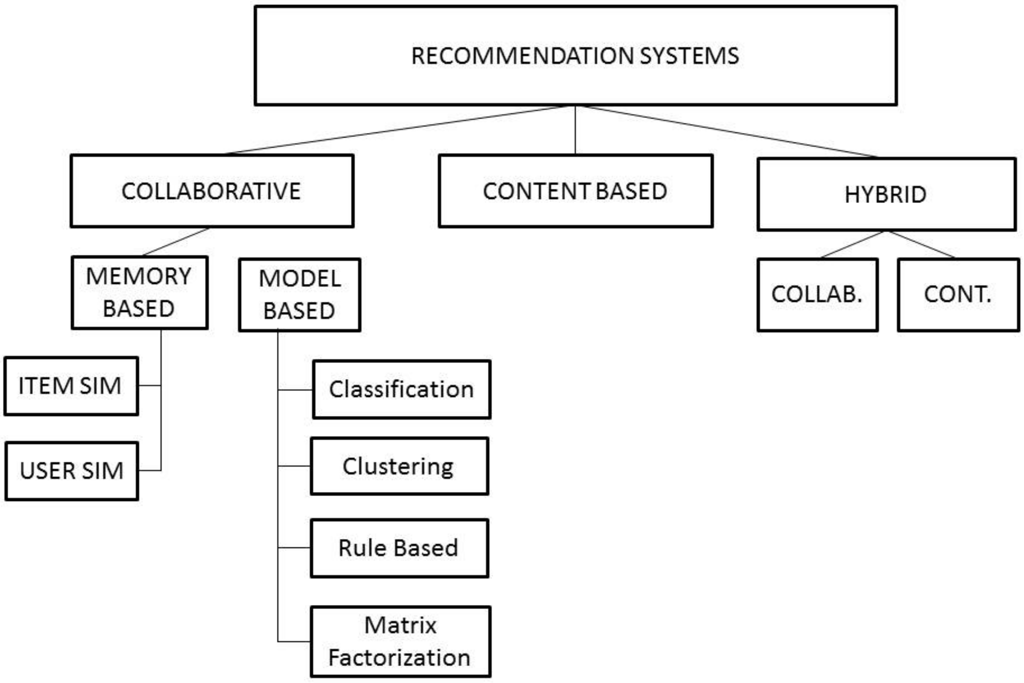

Figure 1.1. Hierarchical representation of the methodologies used in the recommendations systems

Recommendation systems are applied in many different domains such as hotels, restaurants, news, books, social media etc. (Sridevi et al., 2016). Among these, movie recommendation systems are very popular in the literature and have many different examples.

MovieLens (https://grouplens.org) is a well-known movie recommender. It is based on collaborative filtering. Apart from the most of the collaborative approaches that use a single similarity scheme, MovieLens combines both the user and item similarities. It uses matrix factorization to handle the dimensionality problem. Each new user is required to rate a number of movies of different types in the sign up process. In this way, user preferences are learned and the cold start problem avoided. However, reliability of the rating information is questionable because it is provided in the sign up process. An-other drawback of the MovieLens is, biased to movies with highest rates.

MovieREC (Kumar et al., 2015), uses an advanced form of K-Means algorithm. The movies are filtered according to attribute weights before entering the K-Means algorithm. Five attributes are selected and the highest weight is given to the rating information. Clustering algorithm divides the movie space into four clusters. And movies are recommended according to these clusters. There are some drawbacks of this approach. The cluster number (k) is fixed to four. An optimized k number could result in better clusters. The weighting scheme pre filters the movies which reduces the complexity for K-Means. However, it is based on only five attributes. The system can perform better if all at-tributes are considered in the weighting scheme. More importantly, this system is tested on a small dataset and no valid statistical evaluation is made.

Grupta (2015) proposed a collaborative filtering approach based on hierarchical clustering. In this work, users are divided into clusters according to hierarchical clustering and recommendation is done by cluster voting. Chameleon is chosen over K-Means for hierarchical clustering. Rating recommendations are evaluated by mean square error. Although optimization for hierarchical clustering and voting scheme is conducted, generalization of the system over different movie datasets raises some questions. The hierarchical clustering algorithm is mostly based on relative interconnectivity. Single, average and complete linkage must be considered for interconnectivity. Intra cluster similarity must also be calculated to identify the similarity of the items in a specific cluster. If this similarity is low that means clusters are poorly formed and recommendations cannot be trusted. The algorithm uses 25 partition points to divide or combine the existing clusters. These points are data dependent and can be high or low for the smaller or larger datasets. Instead of a fixed partition number, a dynamic optimization based on data volume could be conducted.

Content based filtering proposes a more personalized approach. Rather than similar users, preferences of a specific user are used for the recommendation. For this aim, a user profile vector is constructed. In user profile vector covers the activities of the user over a specific movie (Cami et al., 2017). Bayesian network algorithm is used to form clusters of similar movies. Probability of assigning a movie to a specific cluster is calculated. Then, the recommendation list is formed by the conditional probability calculations. Time and complexity of the probability calculations are the main disadvantages of this method.

When the works above and in the literature are considered we see that both content and collaborative approaches have advantages and disadvantages of their own. Collaborative filtering generalizes the preferences of the similar users. Thus, recommendation is easy for a new user when compared to content based filtering. Content based filtering is more personalized and the preferences of a new user are not known beforehand. New items in collaborative filtering present a similar disadvantage. A new item is not rated yet, thus cannot be recommended even it matches perfectly to a specific user group. These advantages and disadvantages are listed in the Table 1.1.

Table 1.1. Advantages and disadvantages of the filtering approaches in recommendation systems

Approach	Advantage	Disadvantage
Content Based	Personalized Accurate	Always recommend similar things Cannot recommend to a new user Content Problems (Limited, Over-Specialized) Cold-Start
Collaborative	New users are handled easily No need to store content Info	No Content Information Not Personalized Difficult to recommend a new item Not Personalized Items >> users: not enough ratings (Sparsity, Scalability)

In order to combine the advantages and to overcome some of the disadvantages, hybrid methods that combine collaborative and content based filtering are used. Virk et al. (2015) is an example of such system. In this work, the users are required to rate some movies beforehand as in the MovieLens algorithm. This pre-rating information is used to identify the user preferences and grouping the similar users. The recommendation phase is mainly based on collaborative filtering. Content based approach is used to construct a movie and user database but not used for the recommendations. Thus, the system could be considered a collaborative recommender instead of a hybrid.

Tuysuzoglu (2018) proposed a graph based hybrid recommender system. This work combines the collaborative filtering with graph theory. Movies and users are represented as verticals and edges represent the ratings. Statistical calculations are done list the most viewed movie, genre, etc. Unfortunately the effect of using graph theory over the recommendation performance is not given.

Rombouts and Verhoef (2002) designed a hybrid model that uses collaborative and content based filtering. Content based filtering uses a Naïve Bayes model that classifies user ratings. This model is constructed for every user. Prior probability calculations are based on Laplace smoothing. Then these models are linearly combined. They use a weight scheme in order to detect correlations in the collaborative phase. In our opinion, this work has two main drawbacks. One, is the number of Naïve Bayes model is directly proportional with the number of users. That means, the more users are the more calculations. That is why the authors cannot calculate the actual performance of the system for the whole dataset they used. The second drawback is the usage of Laplace smoothing, which can cause bias in the learning data.

Christakou and Stafylopatis (2005) developed a neural network based hybrid recommender. Only three movie attributes, kind, stars and synopsis are taken into consideration. For each user, three neural networks are constructed for these movie attributes. They use only training, test and validation datasets and cross validation is not used, and the performance of the system is high with 82%. For larger datasets the system will suffer from model complexity because for each specific user three different neural networks are constructed.

When we look at the hybrid system examples in the literature, there are two main approaches. In the first one, a basic filtering method is combined with a model-based method, which is mainly based on machine learning. This approach can be considered as model-based filtering rather than a true hybrid system. In the second approach, content based and collaborative filtering methods are combined.

When the researches are examined, we observed that the main shortcoming of the systems using machine learning in the suggestion step is its complexity. The main reason for this complexity is that, the respective hybrid systems develop a specific decision-making model for each user or each product. In order to solve the related problem, collaborative and content based filtering methods were combined in a mixed approach (Burke, 2002) and the multi-layered neural network model was used in the suggestion step. By combining both filtering methods with mixed approach, content based filtering's cold start problem and collaborative filtering's data sparsity problem are solved. When a new user joins the system, his / her individual preferences are not known (cold start), but are assigned to a similar user group through collaborative filtering. Suggestions are made to the new user according to the preferences of this user group. On the other hand, rarely rated items or items that have not yet been rated can be recommended to users with content based filtering. The hybrid model is named as MovieANN, and given in the Figure 1.2.

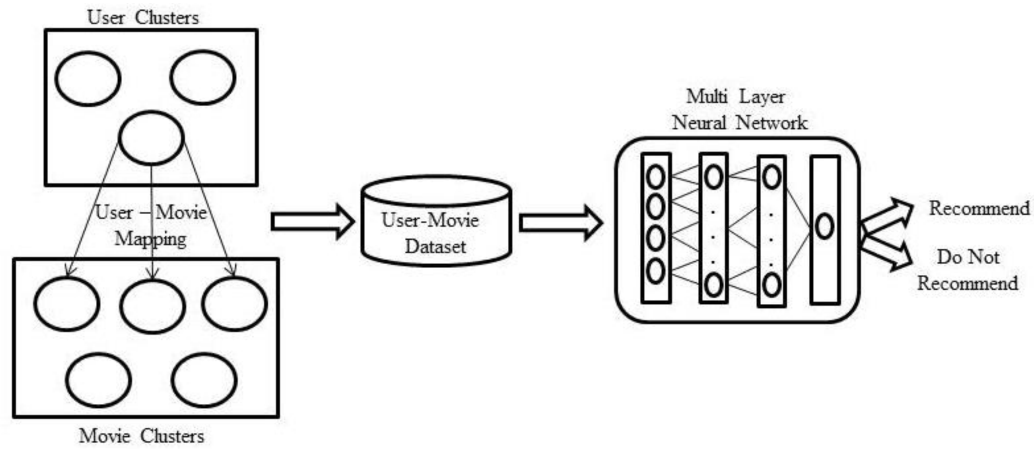

Figure 1.2. Overall structure of the MovieANN

In this system, users and movies are clustered according to the selected attributes. Recommendation step starts with mapping the user cluster into movie clusters. The data set resulting from this mapping is used in the training of artificial neural network model. The relevant model classifies whether a new film will be liked by the user.

The flow of the paper is organized as follows: Section-2 gives a brief background for filtering approaches, Section-3 presents the materials, details of the MovieANN and results are given in the Section-4 and the paper concludes in Section-5.

2. Methodology Background

In this section, a brief methodology background covering collaborative and content based filtering is given.

2.1. Collaborative Filtering

Collaborative filtering is widely used in recommendation systems (Chen et al., 2018). Collaborative filtering methodologies are based on the user – item matrix. This matrix stores the rating information of each item for each user. This matrix is given in the Figure 2.1.

	P_1	P_2	P ...	P_N
U_1	$R_{1\text{-}1}$	$R_{1\text{-}2}$	$R_{1\text{-}}$ ...	$R_{1\text{-}N}$
U_2	$R_{2\text{-}1}$	$R_{2\text{-}2}$	$R_{2\text{-}}$ ...	$R_{2\text{-}N}$
$U_{...}$	$R_{...\text{-}1}$	$R_{...\text{-}2}$	$R_{...\text{-}}$ ...	$R_{...\text{-}N}$
U_M	$R_{M\text{-}1}$	$R_{M\text{-}2}$	$R_{M\text{-}...}$	$R_{M\text{-}N}$

Figure 2.1. The user – item matrix, P indicates products, U indicates users and R indicates the rating given by a user for a specific product

This matrix is formed by the past actions of the user thus, in some studies this collaborative approach is named as Memory Based Collaborative Filtering (Chen et al., 2018]. Based on item – user matrix, recommendations are given according to the similarity of the users or similarity of the items. The user based approach try to group the users with similar preferences on a specific domain. If a product is ranked high among most of the users in the group, it is recommended to the users who did not rank the product yet. Item based approach focuses on the item ratings. For a specific user, rank of a new item is calculated according to its similarity to previously ranked items. In either case, the similarities must be calculated, neighborhood must be formed, and ratings must be assigned.

Cosine similarity (Vit, 2018), Euclidian distance (Draisma, et al., 2014), Pearson correlation (Pearson, 1895), and Jaccard similarity (Michael, 1971) are used to can be used to calculate the similarity of the products or the users. Among them cosine similarity and Pearson correlation is widely used (Bobadilla et al., 2013).

Suppose that there are two users (u_1, u_2) and ***n*** products where P = {p_1, p_2, …, p_n}. The rating vector of the products is given R = {r_1, r_2, …, r_n}. Then the rating vector for u_1 is defined as r_{u1} = {r_{u1p1}, r_{u1p2}, …, r_{u1pn}} and rating vector for u_2 is defined as r_{u2} = {r_{u2p1}, r_{u2p2}, …, r_{u2pn}}. In this condition, similarity between rating vectors r_{u1} and r_{u2} can be calculated by cosine similarity as follows:

$$\cos(r_{u1}, r_{u2}) \frac{\sum_{i=1}^{n} r_{u1pi} . r_{u2pi}}{\sqrt{\sum_{i=1}^{n}(r_{u1pi})^2}\sqrt{\sum_{i=1}^{n}(r_{u2pi})^2}} \tag{2.1}$$

Pearson correlation is the sum of dot products of the difference between specific product rating and average rating of the products (, divided by their sum of root product. More formally:

$$sim(r_{u1}, r_{u2}) \frac{\sum_{i=1}^{n}(r_{u1pi}-\overline{r_{u1}}) . (r_{u2pi}-\overline{r_{u2}})}{\sqrt{\sum_{i=1}^{n}(r_{u1pi}-\overline{r_{u1}})} . \sqrt{\sum_{i=1}^{n}(r_{u2pi}-\overline{r_{u2}})}} \tag{2.2}$$

Neighborhood generation is generally based on K – Nearest Neighbor (KNN) algorithm (Portugal et al., 2018). This algorithm calculates the distance between the active user (u_A) and other users (u). Then according to predefined number ***k***, the nearest ***k*** users form the neighborhood of the active user. Different metrics, such as Euclidian distance, Manhattan distance and Minkowski distance, are used for distance calculations.

$$Euclidian\ (u_A, u) = \sqrt{\sum_{i=1}^{n}(u_A - u_i)^2} \tag{2.3}$$

$$Manhattan\ (u_A, u) = \sum_{i=1}^{n}|u_A - u_i| \tag{2.4}$$

$$Minkowski\ (u_A, u) = (\sum_{i=1}^{n}|u_A - u_i|^x)^{1/x} \tag{2.5}$$

After the user similarities and neighborhood is determined, the rating prediction of the active user (u_A) for a new item, ***i***, is calculated according to the formula (Chen et al., 2018; Resnicket al., 1994) below:

$$r_{u_{A_i}} = \bar{r}_{u_A} + \frac{\sum_{u \in N} sim(u_A, u)\ (r_{ui} - \overline{r_u})}{\sum_{u \in N}|sim(u_A, u)|} \tag{2.6}$$

Instead of similarity, neighborhood and rating prediction calculations, some collaborative filtering approaches use machine-learning methods to construct a decision model (Mahadevan, 2016). In this scheme the model learns the preferences of the similar users for a specific product. The learning scheme can be based on classification, clustering and matrix factorization (Xiao, 2019).

2.2. Content Based Filtering

This filtering method is widely used for text retrieval, news, book or other textual data recommendation (Pazzani, 2007). The methodology identifies the content of the item and this content is matched to user's profile. If the user is interested in the same or similar content, the item is recommended to the user. This scheme requires two important steps. To identify the content of a given item and to form a user profile.

Items are represented as vectors, I = (a_1, a_2, …, a_n) composed of ***n*** attributes. The goal is finding a function F(I) that classifies the input vector. In a real world scenario the importance of the attributes are not equal. Thus, weights are assigned to attributes to indicate the importance. The weights are determined by using TF-IDF (Salton, 1975). If many items contain the attribute a_1, then this attribute could be important and its importance is calculated by TF (term frequency). But high frequency attributes may not be important when differentiating items from each other. Low frequency attributes may be more important for distinguishing the items. This is represented by IDF (inverse term frequency).

Suppose that document set ***D*** consists of ***n*** documents D = {d_1, d_2, …, d_n} and terms are given in the term set T={t_1, t_2, …, t_m }, then TF for term t_1 in d_1 is calculated as follows:

$$TF(t_1, d_1) = \sum_{i=1}^{m} \frac{t_1 \in d_1}{ti \in d_1} \tag{2.7}$$

$$IDF(t_1, D) = \log \sum_{i=1}^{n} \frac{d_i}{t_1 \in d_i} \tag{2.8}$$

Then weight of the term t_1 is calculated as:

$$W_{t_1} = TF_{t_1} * IDF_{t_1} \tag{2.9}$$

These weights are used to construct the content vectors. These vectors are then used to learn the user preferences. Similarity calculations (Pazzani, 1999) and/or model based approaches (Campos et al., 2010) can be used in this learning process.

3. Materials

In order to test the performance of the MovieANN, MovieLens 1M Dataset (Maxwell, 2015) is used. This dataset includes one million ratings of six thousand users for four thousand movies. This data is organized as three separate ".dat" files that are "users.dat", "movies.dat", and "ratings.dat".

Users.dat" file stores user information. The attributes are id, age, sex, occupation and time stamp. The sex attribute is binary, F indicates female and M indicates male. The attributes age and occupation are discrete. The age attribute has seven groups and the occupation attribute has twenty groups. Coding of these groups is given in the Table 3.1 and Table 3.2.

Table 3.1. Coding scheme for the occupation attribute

Group Code	Data
1	academic/educator
2	artist
3	clerical/admin
4	college/grad student
5	customer service
6	doctor/health care
7	executive/managerial
8	farmer
9	homemaker
10	K-12 student
11	lawyer
12	programmer
13	retired
14	sales/marketing
15	scientist
16	self-employed
17	technician/engineer
18	tradesman/craftsman
19	unemployed
20	writer

Table 3.2. Coding scheme for the age attribute

Group Code	Data Range
1	<18
18	18-24
25	25-34
35	35-44
45	45-49
50	50-55
56	>=56

“Movies.dat” file stores information about the movies. The attributes are movie id, movie name and genres. Genres are divided into eighteen subgroups. One movie can have more than one genre. The movie genre coding and the number of movies of a specific genre are given in the Table 3.3.

Table 3.3. Coding scheme for the genre attribute and number of the movies for each specific genre

Genre	Coding	Number of Movies
Action	1	503
Adventure	2	283
Animation	3	105
Children's	4	252
Comedy	5	1200
Crime	6	211
Documentary	7	127
Drama	8	1602
Fantasy	9	68
Film-Noir	10	44
Horror	11	343
Musical	12	114
Mystery	13	106
Romance	14	471
Sci-Fi	15	276
Thriller	16	492
War	17	143
Western	18	68

“Ratings.dat” file stores rating information. The attributes are user id, movie id, rating score and time stamp. The rating score is given as five star scale, with five indicating the highest rating.

4. The MovieANN

MovieANN is a hybrid movie recommendation system that combines collaborative and content based filtering and uses a multi-layer artificial neural network in the recommendation phase.

MovieANN is a hybrid movie recommendation system that combines collaborative and content based filtering and uses a multi-layer artificial neural network in the recommendation phase.

To handle the cold start problem in the content based approach, users are grouped according to their similarities. By this way, a new user is assigned to a specific user group with similar preferences. To handle the data sparsity problem in the collaborative filtering, a content based approach is used. Movies are grouped according to their content similarity. Thus, movies with little or no rating information can also be recommended.
In order to group the similar users and similar movies, clustering is used. For clustering, K-Means (Zahra et al, 2015) and X–Means (Pelleg, 2000) algorithms are preferred. The optimum number for k in the K-Means algorithm is determined by the assumption $k \ll \sqrt{N}$ where N is the total number of instances in the data set. X-Means calculates the optimum k according to Bayesian Information Criteria (Schwarz, 1978). The cluster performances are measured by average intra cluster distance (AID) and Davies-Bouldin Index (DBI) (Davies, 1979). Average intra cluster distance identifies the similarity of the item in a

specific class. DBI gives a ratio of inter cluster dissimilarity and intra cluster similarity. Rapidminer 9.1 (Mierswa, 2019) tool is used for implementing the clustering algorithms.
For movie clustering both K-Means and X-Means divide the movie dataset into two clusters. The AID and DBI are also the same for both methods and calculated as 0.247 and 0.495 respectively. Only the number of movies in the clusters slightly differs. The Figure 4.1 represents the clusters formed by K-Means and X-Means for movie dataset.

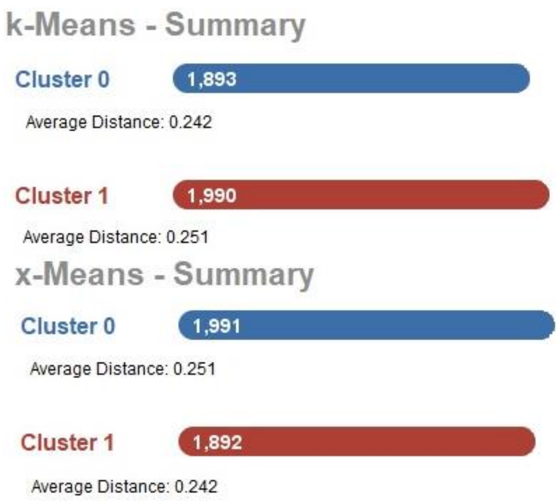

Figure 4.1. Cluster results for movie dataset

To identify the similarity of the users, a user cluster is formed. This cluster is based on the attributes of the "user.dat" dataset, except the timestamp attribute. Timestamp can be used to find similar users. For example, users with an occupation can watch movies at night, users with no occupation can watch movies at daytime. Unfortunately, we cannot infer such information from the timestamp attribute in the data file.

For clustering of the users, K-Means and X-Means give different results. K-Means algorithm divides the users into two clusters while X–Means divides them into four clusters. The results are compared in the Figure 4.2.

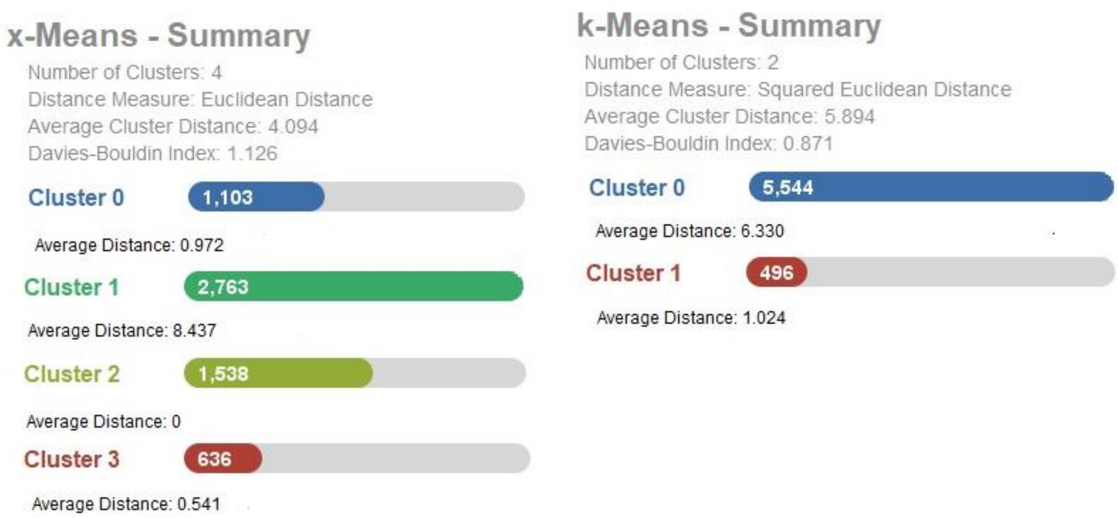

Figure 4.2. Cluster results for user dataset

Figure-4.2 shows that K-Means algorithm gives better results in terms of DBI. However, the clusters are heterogeneous and majority of the data is grouped in the first cluster (Cluster 0). X-Means formed four clusters and they are more homogenous when compared to clusters of K-Means. This condition is verified by the AID. X-Means algorithm gives better results in terms of AID. We can say that clusters of K-Means are distant but the data in a given cluster are also distant. Contrary, in X-Means clusters are closer. Moreover, in each specific cluster, data points are more similar.

In the recommendation phase, a movie is recommended to a user. In this step, the specific cluster, cluster-X, which user belongs is chosen. Movies rated by the users in Cluster-X are retrieved and mapped to users. The user-movie mapping clusters and movie genre distribution among these clusters are given in the Figure 4.3.

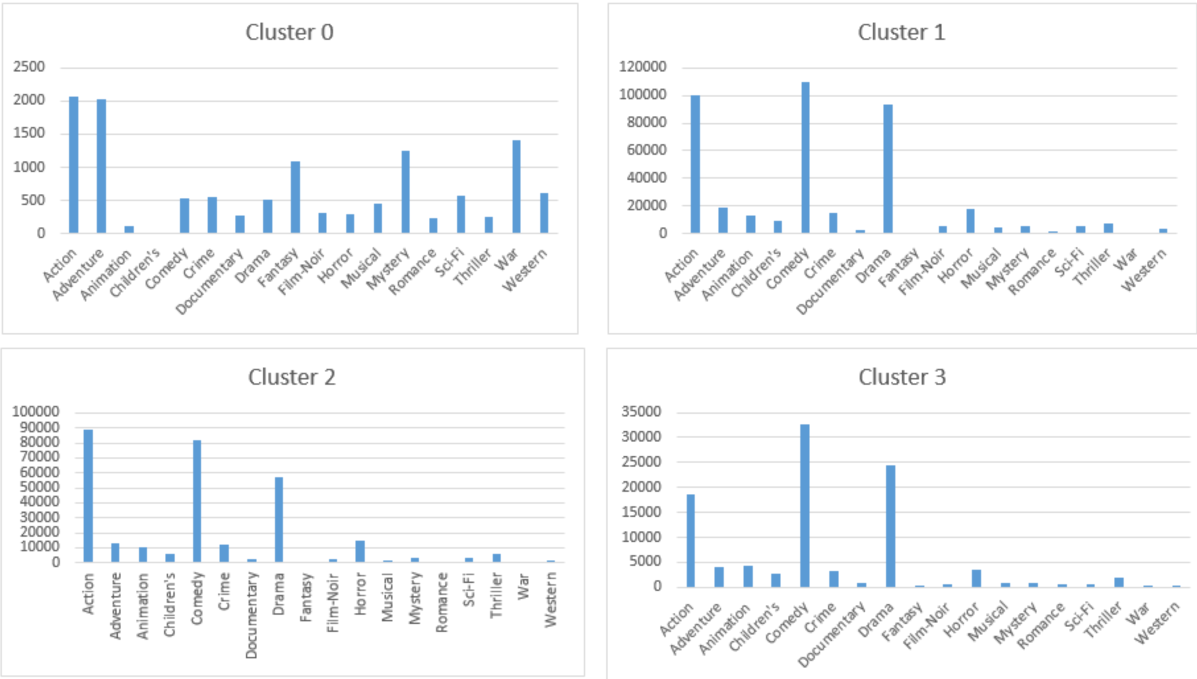

Figure 4.3. User – Movie mapped clusters and distribution of movie genre among these clusters

According to Figue 4.3 the most rated movie genres in Cluster 0 is action and adventure. The users in Cluster 1 mostly rated comedy, action and drama. In Cluster 2, action movies are mostly preferred and followed by comedy and drama.

After the movie-user mapping, a multi-layer artificial neural network model is built over this mapped dataset. The neural network architecture has two hidden layers. The first hidden layer consists of ten nodes and the second hidden layer consists of five nodes. The sigmoid activation function is used in each layer. The model is implemented in Rapidminer 9.3. Learning rate, training cycles and momentum parameters are optimized. Each value range is optimized by step size of four and this gives 125 combinations to test. The value ranges, and the optimum values are given in the Table 4.1.

Table 4.1. The performance results of the recommendation models for each specific cluster

	Min. Value	Max. Value	Optimum Value Cluster-0	Opt. Val. Cluster-1	Opt. Val. Cluster-2	Opt. Val. Cluster-3
Training Cycle	50	200	200	200	200	200
Learning Rate	0	1	0.25	0.25	0.25	0.25
Momentum	0	1	0.333	0.25	0.25	0.25

Optimum training cycles, momentum and learning rates are used for each specific cluster. The multi-layer models for each cluster are given in the Figure 4.4.

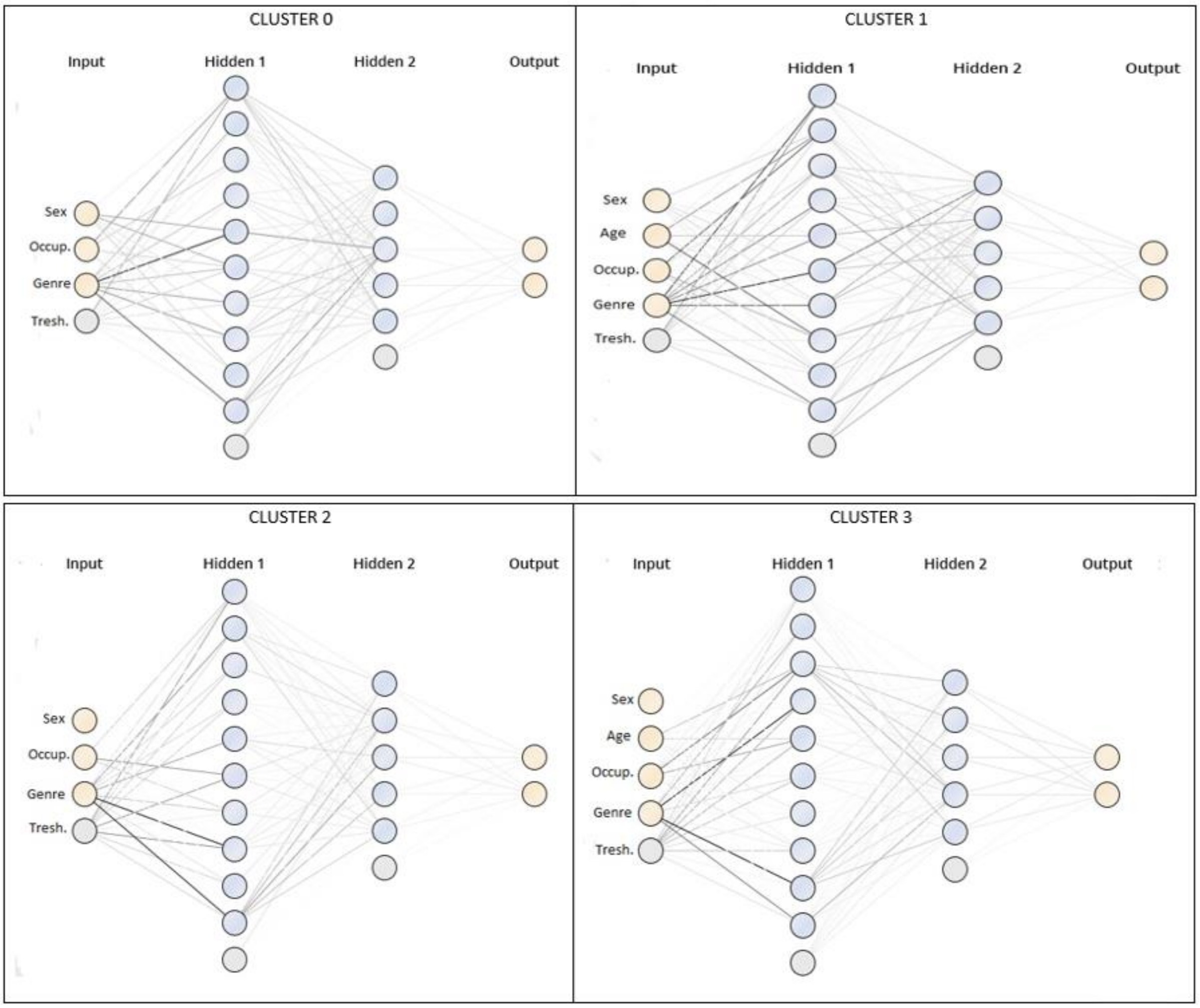

Figure 4.4. Multi-layer neural network structures for each given cluster

As given in the Figure 4.4, age attribute is not given as an input for the clusters 0 and 1. This is because a single age range represents each cluster. On the other hand, in the clusters 1 and 3 the age attribute is given as an input because different age ranges exits in these clusters. The details of each architecture are given in the supplementary material.

For each recommendation model, the data set is divided as 70% training 30% testing. The performances of these models are measured in terms of accuracy, precision and recall. The performance results are given in the Table 4.2.

Table 4.2. The performance results of the recommendation models for each specific cluster

	Model-1 (Cluster 0)	Model-2 (Cluster 1)	Model-3 (Cluster 2)	Model-4 (Cluster 3)
Accuracy	80.86	85.80	82.09	89.33
Precision	80.86	85.87	82.09	89.33
Recall	100	99.9	100	100

The Table 4.2 shows that for all clusters, the recommended movies are relevant in terms of accuracy, precision and recall. The overall accuracy of the MovieANN is 84.52%. In terms of precision and re-call, the overall performance is 84.54% and 99.98% respectively.
In order to compare K-Means and X-Means clusters, the same ANN approach is used. K-Means have two clusters as given in the Figure 4.2. Thus, two separate ANN models are formed. The performance indicators of these models are given in the Table-4.3.

Table 4.3. The performance results of the recommendation models for K-Means clusters

	Model-1 (Cluster 0)	Model-2 (Cluster 1)
Accuracy	83.2	87.92
Precision	100	80
Recall	~0	2.68

As given in the Table 4.3, the decision models formed by K-Means clusters suffer from low recall rates. Low AUC results, 0.588 and 0.621 also consolidate this. The decision models formed by X-Means cluster are more solid and overall performance indicators, including recall, are more promising and higher.

5. Conclusion

Recommendation systems are used to filter the vast amount of data and present relevant information according to the user preferences. These systems can be applied to many different domains and movie recommendation is among them. Movie recommendation systems use three main approaches. These are collaborative filtering, content based filtering and hybrid systems. Collaborative filtering mainly suffers from data sparsity and content based filtering suffers from cold start problem. In order to overcome the disadvantages hybrid systems are proposed. These systems combine the two main filtering approaches by different methodologies such as weighting or augmentation. Some hybrid systems combine a machine learning model along with a filtering method. In such systems, the main problem is the number of decision models. Often, a decision model is built for every specific user or item. This increases complexity as well as the computation time.

In order to overcome the cold start problem of content based filtering and data sparsity of collaborative filtering, we built a hybrid model, MovieANN, based on multi-layer artificial neural network. This model constructs movie and user clusters. Movie clusters are formed in a content based manner and user clusters are formed in a collaborative manner. K-means and X-Means algorithms are used in clustering and final clusters are chosen according to Davies-Bouldin Index and average distance measures. In the recommendation phase, the user and movie clusters are mapped. For a specific user, the cluster of that user is chosen and recommendation model is based on this specific cluster. The recommendation model is based on multi-layer neural network. The network consists of two hidden layers.

MovieANN is tested on MovieLens' 1M dataset and overall accuracy is 84.52%. With overall precision of 84.54% and overall recall of 99.98%, the performance indicators of MovieANN are very high.

6. Future Work

Although MovieANN's recommendation performance is high, we will make improvements to increase the performance even further.

Both user and movie attributes are very limited, depending on the data set used. A parser will be written to retrieve information from movie sites. These information will be processed with TF * IDF method in accordance with the content based approach. This allows more attributes to be processed for the movies.

References

Attarde D.V., Singh M., 2017. Survey on Recommendation System Using Data Mining and Clustering Techniques. International Journal for Research in Engineering Application and Management (IJREAM), 3(9). ISSN : 2454-9150.

Bobadilla J., Ortega F., Hernando A., Gutiérrez A., 2013. Recommender Systems Survey. Knowledge-Based Systems, 46: 109–132.

Burke R., 2002. Hybrid recommender systems: Survey and experiments. User Model User Adapt Interact, 12: 331–370.

Cami B. R., Hassanpour H., Mashayekhi H., 2017. A Content-Based Movie Recommender System Based on Temporal User Preferences. Third Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS). DOI: 10.1109/ICSPIS.2017.8311601.

Campos L.M, Fernández-Luna J.M., Huete J.F., Rueda-Morales M.A., 2010. Combining Content-Based and Collaborative Recommendations: A Hybrid Approach Based on Bayesian Networks. International Journal of Approximate Reasoning, 51: 785–799.

Chen R., Hua Q., Chang Y.S., Wang B., Zhang L., Kong X.A, 2018. Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based On Social Networks. IEE Access. DOI: 10.1109/ACCESS.2018.2877208.

Christakou C., Stafylopatis A., 2005. Hybrid Movie Recommender System Based on Neural Networks. Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA'05). DOI: 10.1109/ISDA.2005.9.

David D.L., Bouldin D.W., 1979. A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-1(2): 224–227. DOI:10.1109/TPAMI.1979.4766909

Draisma J., Horobeţ E., Ottaviani G., Sturmfels B., Thomas R., 2014. The Euclidean Distance Degree. arXiv:1309.0049.

Gupta U., Patil N., 2015. Recommender System Based On Hierarchical Clustering Algorithm Chameleon. IEEE International Advance Computing Conference (IACC). DOI: 10.1109/IADCC.2015.7154856.

Harper M.F., Konstan J.A., 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS), 5(4).

Haruna K., Ismail M. A., Damiasih D., Sutopo J., 2017. A Collaborative Approach for Research Paper. PloS One, 12(10): 1-17.

Karimi, M., Jannach, D. & Jugovac, M.(2018). News recommender systems – Survey and roads ahead Information Processing & Management, 54(6): 1203-1227.

Koohi H., Kiani K., 2017. A New Method To Find Neighbor Users That Improves The Performance Of Collaborative Filtering. Expert Systems with Applications: An International Journal, 83(C): 30-39.

Kumar M., Yadav D.K., Singh A., Gupta V.K., 2015. A Movie Recommender System: MOVREC. International Journal of Computer Applications, 124(3).

Lekakos G., Caravelas, P., 2008. A Hybrid Approach for Movie Recommendation. Multimed Tools Appl, 36: 55–70.

Levandowsky M., David W., 1971. Distance Between Sets. Nature, 234(5): 34–35.

Mahadevan A., Arock M., 2016. A Study and Analysis of Collaborative Filtering Algorithms for Recommender Systems. International Journal of Circuit Theory and Applications, 9(27): 127-136.

Mierswa I., Klinkenberg R., 2019. Rapidminer Studio 9.1: Data Science, Machine Learning, Predictive Analytics.

Pazzani M.J., 1999. A Framework for Collaborative, Content-Based and Demographic Filtering. Artificial Intelligence Review, 13: 393–408.

Pazzani M.J., Billsus D., 2007. Content-Based Recommendation Systems. In: Brusilovsky P., Kobsa A., Nejdl W. Eds. The Adaptive Web. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. 4321.

Pearson K., 1895. Notes on Regression and Inheritance in the Case of Two Parents. Proceedings of the Royal Society of London, 58: 240–242.

Pelleg D., Moore A., 2000. X-Means: Extending K-Means with Efficient Estimation of The Number Of Clusters. In Proceedings of the 17th International Conf. on Machine Learning, 727 – 734.

Portugal I., Alencar P., Cowan D., 2018. The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review. Expert Systems with Applications, 97: 205 – 227.

Resnick P., Iacovou N., Suchak M., Bergstrom P., Riedl J., 1994. Grouplens: An Open Architecture for Collaborative Filtering Of Netnews. In Proceedings of ACM Conference on Computer Supported Cooperative Work. DOI: 10.1145/192844.192905.

Rombouts J., Verhoef T., (Date of access: July 2019). A Simple Hybrid Movie Recommender System. http://www.fon.hum.uva.nl/tessa/Verhoef/Past_projects_files/Eind_Rombouts_Verhoef.pdf.

Salton G., Wong A., Yang C.S., 1975. A Vector Space Model for Automatic Indexing. Communications of the ACM, 18(11).

Sridevi M., Rao R.R., Rao M.V., 2016. A survey on recommender system. International Journal of Computer Science and Information mSecurity (IJCSIS), 14(5).

Schwarz G.E., 1978. Estimating the Dimension of a Model. Annals of Statistics, 6(2): 461–464.

Tüysüzoğlu G., Işık Z., 2018. Hybrid Movie Recommendation System Using Graph-Based Approach. International Journal of Computing Academic Research (IJCAR), 7(2): 29-37.

Virk H.K., Singh M., Singh A., 2015. Analysis and Design of Hybrid Online Movie Recommender System. IJIET 5(2).

Vít N., 2018. Implementation Notes for The Soft Cosine Measure. In Proceedings The 27th ACM International Conference On Information And Knowledge Management, 1639–1642.

Xiao T., Shen H., 2019. Neural Variational Matrix Factorization with Side Information for Collaborative Filtering. In: Yang Q., Zhou Z.H., Gong Z., Zhang M.L., Huang S.J., Eds. Advances in Knowledge Discovery and Data Mining. PAKDD 2019. Lecture Notes in Computer Science. Springer, Cham.

Zahra S., Ghazanfar M.A., Khalid A., Azam M.A., Naeem U., Bennett A.P., 2015. Novel Centroid Selection Approaches for K-Means Clustering Based Recommender Systems. Information Sciences, 320(1): 156-189.