Movie Recommendation System Based on Fuzzy Inference System and Adaptive Neuro Fuzzy Inference System

Mahfuzur Rahman Siddiquee, Electrical and Computer Engineering Department, North South University, Dhaka, Bangladesh

Naimul Haider, Electrical and Computer Engineering Department, North South University, Dhaka, Bangladesh

Rashedur M. Rahman, Electrical and Computer Engineering Department, North South University, Dhaka, Bangladesh

ABSTRACT

One of most prominent features that social networks or e-commerce sites now provide is recommendation of items. However, the recommendation task is challenging as high degree of accuracy is required. This paper analyzes the improvement in recommendation of movies using Fuzzy Inference System (FIS) and Adaptive Neuro Fuzzy Inference System (ANFIS). Two similarity measures have been used: one by taking account similar users' choice and the other by matching genres of similar movies rated by the user. For similarity calculation, four different techniques, namely Euclidean Distance, Manhattan Distance, Pearson Coefficient and Cosine Similarity are used. FIS and ANFIS system are used in decision making. The experiments have been carried out on Movie Lens dataset and a comparative performance analysis has been reported. Experimental results demonstrate that ANFIS outperforms FIS in most of the cases when Pearson Correlation metric is used for similarity calculation.

Keywords: *ANFIS, Collaborative Filtering, Fuzzy Logic, Movie Recommender System, Neural Network*

1. INTRODUCTION

Item recommendation is popularly known for its use in most social networks and e-commerce websites where the customer's choice information is analyzed to generate a list of most likely products which the customer will be interested in. Different recommender systems use different techniques to perform these recommendations. One of the most common techniques is to analyze the customer's previous purchase records. Once a recommender system is designed, any product can be recommended using the system. In our system, we build a system for recommending

DOI: 10.4018/IJFSA.2015100103

movies. Generally movies are recommended using two techniques, collaborative filtering and content-based filtering.

One of the most important and effective technique is collaborative filtering. It works by creating a database for the preferred items by users. A user is compared against the database to find neighbors who have similar taste like him. Items that the neighbors like are then recommended to the user with the expectation that he will also like them. Collaborative filtering has proven to be effective in many areas information filtering applications and e-commerce sites. Yet, there are still challenges and room for improvement in this technique. Challenges include improving scalability since it is required to search millions of data. Since a single user can rate many items, there are thousands of data points to be considered. Another challenge is to improve the quality of recommendations to the users. Users require recommendations that help them find the items they like. It is important to consider both the challenges simultaneously in order to retain the performance since the less time an algorithm takes to search for neighbors, the more it becomes scalable. In this paper, we compare the results using different techniques to measure similarity and analyze the performance of the techniques.

Our objective is to recommend movies to a user. Let us rephrase this as if we would like to find the expected rating a user might give to a movie. By looking at the predicted rating, we can judge whether the user may like the movie or not.

Suppose we are recommending a target movie to a target user. At first, we define the term neighborhood as the users who have similar choice to the target user. By similar choice, we mean the user's similarity in rating to the movies which other users also rated. To calculate choice similarity, we used Euclidean Distance, Manhattan Distance, Pearson Correlation Coefficient and Cosine Similarity. Then we took the K most similar users and found the average rating they have given to the target movie. This is our first input parameter. Secondly, we need to find similar movies to the target movie. To do that, we consider the movies that have the highest number of matching genres and we also found this using Euclidean Distance, Manhattan Distance, Pearson Correlation Coefficient and Cosine Similarity. Then we took the K most similar movies and calculated the average rating the target user has given to these movies. This is our second input parameter which we call Acceptance Rate. Then we calculated expected rating using these two parameters and the decision making is made using Mamdani FIS and ANFIS systems. Then we compared the performance of the systems by varying parameters, value of K, and different membership functions.

2. RELATED WORK

Fuzzy logic has been used in many areas that include recommendation systems, fuzzy controller, robotics etc. Recently, Semwal et al. (2015) designed fuzzy logic controller that can predict push recovery strategy for a robot. Fuzzy rules were defined in terms of roll and pitch to avoid high variability. One of the earliest implementations of recommender systems is Tapestry (Goldberg, Nichols, Oki & Terry, 1992). It depends on the opinions of people on small connected communities like office workgroups, student networks, etc. Recommender systems for large communities cannot be dependent on one another. Many other recommender systems were developed, such as Ringo (Shardanand & Maes, 1995) and Video Recommender (Hill, Stead, Rosenstein & Furnas, 1995). An issue of ACM (Resnick & Varian, 1997) discusses a number of different recommender systems.

Technologies used in recommender systems include Bayesian Networks (Breese, Heckerman & Kadie, 1998), Clustering (Sarwar, Karypis, Konstan & Riedl, 2001) and Horting (He &

Chu, 2010). Bayesian networks (Breese, Heckerman & Kadie, 1998) use training data to create a decision tree where user's information is represented by nodes and edges of the graph. The resultant model is fast and as accurate as nearest neighbor methods. Clustering techniques identify groups with similar interests. The predictions are made by averaging the preferences of the cluster. Clustering (Sarwar, Karypis, Konstan & Riedl, 2001) techniques produce less significant output than nearest neighbor methods. Horting (He & Chu, 2010) is a technique where the nodes are users and the edges are the weight of similarity between users. Similarity predictions are made by traversing the graph to nearest nodes and the accumulating the preference of the nearby users.

In a research (Sarwar, Karypis, Konstan & Riedl, 2001) the authors attempted to improve the methods of collaborative filtering using item based recommendation. This system works by finding the set of similar items for each set of items. Their technique separates the users who have both rated the same item and then apply a similarity computation algorithm to determine the similarity. Their techniques include Cosine-based Similarity, Correlation-based similarity and Adjusted Cosine Similarity. Their research also used the data form MovieLens with 100,000 ratings. According to their performance results, item-based recommendation provides better quality of prediction than user-based algorithm. Their main focus was scalability issues and they proved that their model has performance benefits.

In another research (He & Chu, 2010) the authors looked upon a variety of methods in designing recommender system such as Immediate Friend Inference, User Preference, Item Acceptance, Influence from Immediate Friends and Distant Friend Inference. They used the dataset of YELP which is a social site recommending restaurants and locations. They experimented on several different methods to reach their conclusion. Methods include:

I. Friend Average: Predicts the ratings of the target users on target items with the average ratings of their immediate friends.
II. Weighted Friends: This considers that every immediate friend has a different impact on the target user. The more the impact from an immediate friend, the closer the target user's rating is to that of the friend.
III. Naïve Bayes: They implemented a social form of Bayesian Network using a naïve Bayes classifier.
IV. Collaborative Filtering

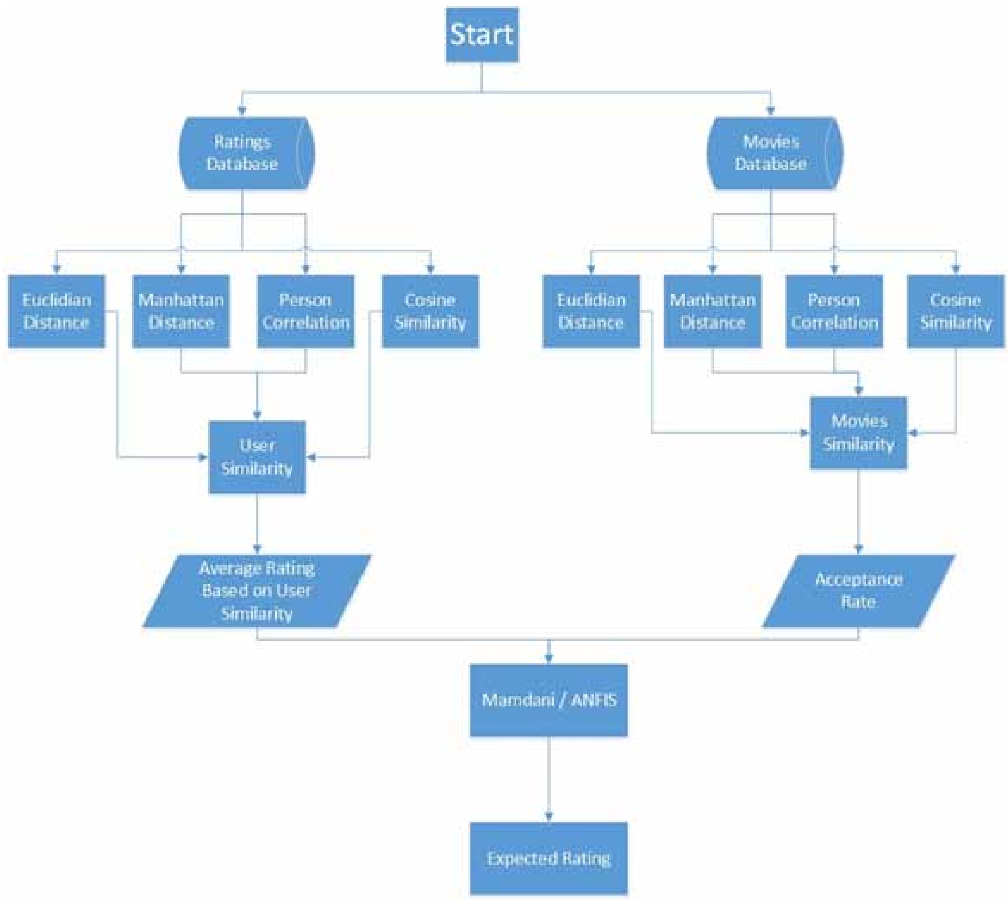
From their research, they concluded that friends have a tendency to review the same restaurants and give similar ratings.

The research (Roy & Kundu, 2013) was motivated by "Netflix Prize Challenge". In the research, the authors used collaborative filtering and data clustering to recommend movies to users. They used the dataset of Netflix. They used Euclidean distance, Manhattan distance and Minkowski distance to measure the similarity between users. They created a test set of 50 users for collaborative filtering. From their research, they have extracted the unique genres from the movie table and then created a user data set which has four randomly selected genres from the available genres of movies and their corresponding normalized genre weights. Movies were recommended to the new user according to his choice of dimensions. Their next aim is to recommend movies to new user based on genre as well as other dimensions like actors, actresses etc.

3. ARCHITECTURE

The general architecture is presented in Figure 1.

Figure 1. Recommender system architecture



3.1. Distance Metric

The primary reason behind using the collaborative filtering approach is to approximate a possible rating a user may give to a movie analyzing the user's previous rating history and the likeliness of the users who are most similar to the user. The technique to find the most similar users is known as K-Neighborhood method. To find the similarity between two users, we have considered four different methods to analyze performance.

3.1.1. Euclidean Distance Measurement

Euclidean Distance is basically the distance between two points. According to Euclidean Distance Measurement the dissimilarity between user u and v is shown in equation 1.

$$D_{uv} = \sqrt{\sum_{i=1}^{n} \left| a_{ui} - a_{vi} \right|^2} \tag{1}$$

Where $u, v \in U$ and $v \neq u$

Here, U is the set of all users. D_{uv} is the distance between user u and v based on their movie ratings. a_{ui} and a_{vi} are ratings of user u and v respectively on movie i.

We also measured dissimilarity among movies based on genres to get a list of similar movies. According to Euclidean Distance the dissimilarity between movie 1 and 2 is calculated as equation 2.

$$D_{12} = \sqrt{\sum_{i=1}^{n} \left| a_{1i} - a_{2i} \right|^2} \tag{2}$$

Here, a_{1i} , a_{2i} are the values of i^{th} genre for movie 1 and movie 2 respectively. If a movie is of i^{th} genre, then this value is 1, or 0 otherwise.

3.1.2. Manhattan Distance Measurement

Manhattan Distance was defined based on grid like street geography. Using Manhattan Distance we measured the dissimilarity between user u and v like equation 3.

$$D_{uv} = \sum_{i=1}^{n} \left| a_{ui} - a_{vi} \right| \tag{3}$$

where $u, v \in U$ and $v \neq u$

Here, U is the set of all users. D_{uv} is the distance between user u and v based on their movie ratings. a_{ui} and a_{vi} are ratings of user u and v respectively on movie i.

We also measured movie dissimilarity using Manhattan Distance. Dissimilarity between two movies 1 and 2 is represented in equation 4.

$$D_{12} = \sum_{i=1}^{n} \left| a_{1i} - a_{2i} \right| \tag{4}$$

Here, a_{1i} , a_{2i} are the values of i^{th} genre for movie 1 and movie 2 respectively. If a movie is of i^{th} genre then this value is 1 or 0 otherwise.

3.1.3. Pearson Coefficient Measurement

Manhattan distance and Euclidean distance measurement method has some issues. It is not guaranteed that all users will give rating in the same range, i.e., some users tend to give rating in a narrow band where some others give rating in a wide band. This issue is not handled by Manhattan distance and Euclidean distance. To handle this 'grade inflation' problem we use Pearson Correlation Coefficient method. This method measures similarity on a scale of [-1, 1], where 1 means perfectly similar and -1 means perfectly dissimilar. So the similarity between user *u* and *v* is defined by equation 5.

$$S_{uv} = \frac{\sum_{i=1}^{n}\left(a_{ui} - \underline{a_u}\right)\left(a_{vi} - \underline{a_v}\right)}{\sqrt{\sum_{i=1}^{n}\left(a_{ui} - \underline{a_u}\right)^2}\sqrt{\sum_{i=1}^{n}\left(a_{vi} - \underline{a_v}\right)^2}} \tag{5}$$

where,

$$\underline{a_u} = \frac{\sum_{i=1}^{n} a_{ui}}{n}$$

and

$$\underline{a_v} = \frac{\sum_{i=1}^{n} a_{vi}}{n}$$

Here, a_{ui} and a_{vi} are ratings of user *u* and *v* respectively on movie *i*. $\underline{a_u}$ and $\underline{a_v}$ are average ratings of user *u* and *v* respectively on all movies.

To get similar movie list we also used Pearson Correlation Coefficient. Similarity between two movies 1 and 2 is given by equation 6.

$$S_{12} = \frac{\sum_{i=1}^{n}\left(a_{1i} - \underline{a_1}\right)\left(a_{2i} - \underline{a_2}\right)}{\sqrt{\sum_{i=1}^{n}\left(a_{1i} - \underline{a_1}\right)^2}\sqrt{\sum_{i=1}^{n}\left(a_{2i} - \underline{a_2}\right)^2}} \tag{6}$$

Here, a_{1i}, a_{2i} are the values of i^{th} genre for movie 1 and movie 2 respectively. If a movie is of i^{th} genre then this value is 1, or 0 otherwise. $\underline{a_1}$ and $\underline{a_2}$ are average values of genres of movie 1 and 2.

3.1.4. Cosine Similarity Measurement

This method is based on the dot product of vectors. It basically measures the angle between two vectors and based on that it defines how similar or close two vectors are. Like the Pearson Correlation Coefficient method, Cosine Similarity also measures similarity on a scale of [-1, 1], where 1 means perfectly similar and -1 means perfectly dissimilar. If we define the attribute vector of user *u* as A_u and attribute vector of user *v* as A_v, then Cosine Similarity can be written as equation 7.

$$cos\left(u,v\right)=\frac{A_u.A_v}{\|A_u\|\ \|A_v\|} \tag{7}$$

We used the ratings of each movie as the attribute of these vectors.

Similarly, the similarities of movies are measured using Cosine Similarity. We used genre as the attribute of the vectors. If we define the attribute vector of movie *1* as A_1 and attribute vector of movie *2* as A_2, then Cosine Similarity is given by equation 8.

$$cos\left(1,2\right)=\frac{A_1.A_2}{\|A_1\|\ \|A_2\|} \tag{8}$$

3.2. Expected Rating Calculation

Our technique is similar to the K-Neighborhood method. Suppose we want to recommend a movie to Alice. For all users other than Alice, we find the similarity measurement of them with Alice using all the four techniques. Then we sort the list, and keep K users who are most similar with Alice. We varied the value of K to be 5, 10 and 20.

For example, we want to recommend the movie Toy Story to Alice. We will try to predict the rating that will be given to this movie by Alice. Based on that expected rating value we will prioritize the movies. We mentioned earlier that we will try to find out user's (Alice) choice. This user choice information will be used to calculate the expected rating of the movie Toy Story by Alice.

To get Alice's choice information, we will find a set of users (let us call this neighbors set) with similar types of choice as of Alice. After that, we will calculate the average rating of neighbors to the movie Toy Story. To calculate the expected rating of Alice to the movie Toy Story, we will also consider the acceptance rate of this movie to Alice. To get this acceptance rate we will calculate the average rating of Alice to the movies similar to Toy Story. After getting these two information (Alice's choice information and acceptance rate of Toy Story to Alice), we put these information into our fuzzy inference system to get how significantly the movie Toy Story will be recommended to Alice. We use the defuzzified crisp value of output as the expected rating of Toy Story by Alice. This whole process is shown in Figure 1.

So far we are trying to calculate the expected rating of movie Toy Story by user Alice and we have seen the various ways of calculating one input of our fuzzy inference system 'Average Rating based on User Choice Similarity'. There we used Alice's neighborhood rating to predict her choice. In this part, we will use Alice's previous average rating on movies similar to Toy Story. We call it 'Average Rating based on Acceptance Rate' (A_{ri}). We will use the similar methods for finding similarity between movies those we used to find similarity between users. For the movies we will use genres as attributes. As our dataset contains 18 genres, all the movies will have 18 attributes and thus we will represent our data in 18 dimensions. We will use 0 and 1 to indicate whether a movie is of a particular genre. 1 means yes and 0 means no.

Therefore we have identified two input parameters of our fuzzy inference system:

a. Average rating to target movie by users similar to the target user (Alice).
b. Average rating to movies similar to target movie (Toy Story) by target user (Alice). We denote this as Acceptance Rate.

3.3. Example

3.3.1. User Similarity Calculation

Let us consider 3 users (Alice, Bob, Leonard) who have rated the movies Toy Story, Golden Eye and The Parent Trap as Table 1.

Let us find out the similarity of other users with Alice using different distance metrics.

3.3.1.1. Using Manhattan Distance

According to the equation 3 given by Manhattan Distance method, the distance from Alice to Bob and Leonard is given in Table 2.

Since the distance from Alice to Bob is less than the distance from Alice to Leonard, we can conclude that Bob is more similar with Alice.

3.3.1.2. Using Euclidean Distance

Similarly using Euclidean Distance formula of equation 1, the distance from Alice to Bob and Leonard are calculated in Table 3.

From Table 3, we can see that the distance from Alice to Bob is less that the distance from Alice to Leonard. So Bob is more similar to Alice.

3.3.1.3. Using Pearson Correlation Coefficient

As we mentioned earlier, the choice of two users may be similar but their rating might be different. Pearson Correlation Coefficient can figure out the perfect agreement of users. Similarity of Bob and Leonard with Alice is calculated in Table 4 using equation 5.

In this case, we can see that their choices are actually same.

3.3.1.4. Using Pearson Correlation Coefficient

According to the equation 7 of Cosine Similarity, the similarity calculations are shown in Table 5.

We can notice that Bob is more similar to Alice using this method also.

3.3.2. Movie Similarity Calculation

For movie similarity calculation, let us consider 3 movies (Toy Story, Golden Eye and The Parent Trap). The respective genre representation is given at Table 6. Here 1 means the movie belongs to that particular genre and 0 means otherwise.

3.3.2.1.. Using Manhattan Distance

Using equation 4 of Manhattan Distance to calculate similarity of movies, we can see from Table 7 that the movie “The Parent Trap” is more similar to the movie “Toy Story”.

Table 1. User Ratings

	Toy Story (1995)	Golden Eye (1995)	The Parent Trap (1998)
Alice	5	5	4
Bob	2	5	3
Leonard	1	4	2

Table 2. User similarity based on Manhattan distance

User	Distance from Alice
Bob	4
Leonard	7

Table 3. User similarity based on Euclidean distance

User	Distance from Alice
Bob	3.16
Leonard	4.58

Table 4. User similarity based on Pearson Correlation Coefficient

User	Similarity With Alice
Bob	0.19
Leonard	0.19

Table 5. User Distance using Cosine Similarity

User	Similarity With Alice
Bob	0.94
Leonard	0.89

Table 6. Movie Genre Representation

Movie	Animation	Children's	Comedy	Thriller	Adventure	Action
Toy Story	**1**	**1**	**1**	**0**	**0**	**0**
Golden Eye	**0**	**0**	**0**	**1**	**1**	**1**
The Parent Trap	**0**	**1**	**0**	**0**	**0**	**0**

Table 7. Movie Similarity Using Manhattan Distance

Movie	Distance from Toy Story
Golden Eye	6
The Parent Trap	2

3.3.2.2. Using Euclidean Distance

In case of using the equation 2 of Euclidean Distance to calculate similar movies, we can see from Table 8 that the movie “The Parent Trap” is more similar to the movie “Toy Story” as well.

3.3.2.3. Using Pearson Correlation Coefficient

Using equation 6 of Pearson Correlation Coefficient method Table 9 shows that the movie “Golden Eye” is perfectly dissimilar to the movie “Toy Story”.

Therefore it is shown that the movie “The Parent Trap” is more similar to “Toy Story” also by this calculation.

3.3.2.4. Using Cosine Similarity

Cosine Similarity calculation by equation 8 shows that the movie “Golden Eye” is not similar at all to the movie “Toy Story” in Table 10. The movie “The Parent Trap” is more similar to “Toy Story” than “Golden Eye”.

In this way, we have calculated the inputs of our fuzzy inference system.

Table 8. Movie Similarity Using Euclidean Distance

Movie	Distance from Toy Story
Golden Eye	2.45
The Parent Trap	1.41

Table 9. Movie Similarity Using Pearson Correlation Coefficient

Movie	Similarity with Toy Story
Golden Eye	-1
The Parent Trap	0.45

Table 10. Movie Similarity Using Cosine Similarity

Movie	Similarity with Toy Story
Golden Eye	0
The Parent Trap	0.58

4. FUZZY INFERENCE SYSTEM

Now we have our two inputs 'Average Rating based on User Choice Similarity' (A_{ru}) and 'Average Rating based on Acceptance Rate' (A_{ri}) for fuzzy inference system (FIS). Each of these inputs will have a value from {LOW, MEDIUM, HIGH}. Based on our following rule set FIS will generate output indicating how will be the expected rating. We varied the number of output membership functions to 3, 4, and 5. The value of fuzzy output variable for 3, 4, and 5 membership functions will be from {LOW, MEDIUM, HIGH}, {VERY_LOW, MEDIUM, HIGH}, and {VERY_LOW, LOW, MEDIUM, HIGH, VERY_HIGH} respectively. Table 11, 12, and 13 show rule sets for 3, 4, and 5 output membership functions. The top most row and left most column of these tables represent the value of fuzzy input variables.

Table 11. Rule set of 3 output membership functions

A_{ru} / A_{ri}	**LOW**	**MEDIUM**	**HIGH**
LOW	LOW	LOW	MEDIUM
MEDIUM	LOW	MEDIUM	HIGH
HIGH	MEDIUM	HIGH	HIGH

Table 12. Rule set of 4 output membership functions

A_{ru} / A_{ri}	**LOW**	**MEDIUM**	**HIGH**
LOW	VERY LOW	LOW	MEDIUM
MEDIUM	LOW	MEDIUM	HIGH
HIGH	MEDIUM	HIGH	HIGH

Table 13. Rule set of 5 output membership functions

A_{ru} / A_{ri}	**LOW**	**MEDIUM**	**HIGH**
LOW	VERY LOW	LOW	MEDIUM
MEDIUM	LOW	MEDIUM	HIGH
HIGH	MEDIUM	HIGH	VERY HIGH

4.1. Triangular Membership Functions

Inputs

Average Rating by User (avgRatingByUser): Membership functions of this variable are defined by equation 9, 10, and 11.

$$\mu_{low}(x)=\begin{cases} 0, if\ x<0 \\ \frac{2.5-x}{2.5}, if\ 0\le x\le 2.5 \\ 0, if\ x\ge 2.5 \end{cases} \tag{9}$$

$$\mu_{medium}(x)=\begin{cases} 0, if\ x\le 1 \\ \frac{x-1}{1.5}, if\ 1\le x\le 2.5 \\ \frac{4-x}{1.5}, if\ 2.5\le x\le 4 \\ 0, if\ x\ge 4 \end{cases} \tag{10}$$

$$\mu_{high}(x)=\begin{cases} 0, if\ x<2.5 \\ \frac{x-2.5}{2.5}, if\ 2.5\le x\le 5 \\ 0, if\ x>5 \end{cases} \tag{11}$$

Average Rating to Movie (avgRatingToMovie): This is the second fuzzy input variable and its membership functions are defined by equation 12, 13, and 14.

$$\mu_{low}(x)=\begin{cases} 0, if\ x<0 \\ \frac{2.5-x}{2.5}, if\ 0\le x\le 2.5 \\ 0, if\ x\ge 2.5 \end{cases} \tag{12}$$

$$\mu_{medium}(x)=\begin{cases} 0, if\ x\le 1 \\ \frac{x-1}{1.5}, if\ 1\le x\le 2.5 \\ \frac{4-x}{1.5}, if\ 2.5\le x\le 4 \\ 0, if\ x\ge 4 \end{cases} \tag{13}$$

$$\mu_{high}(x)=\begin{cases} 0, if\ x<2.5 \\ \frac{x-2.5}{2.5}, if\ 2.5\le x\le 5 \\ 0, if\ x>5 \end{cases} \tag{14}$$

Output

Using 3 Output Membership Functions: For 3 output membership functions we defined them as equation 15, 16, and 17.

$$\mu_{low}\left(x\right)=\begin{cases} 0, if\ x<0 \\ \dfrac{2.5-x}{2.5}, if\ 0\le x\le 2.5 \\ 0, if\ x\ge 2.5 \end{cases} \tag{15}$$

$$\mu_{medium}\left(x\right)=\begin{cases} 0,\ if\ x\le 1 \\ \dfrac{x-1}{1.5},\ if\ 1\le x\le 2.5 \\ \dfrac{4-x}{1.5},\ if\ 2.5\le x\le 4 \\ 0,\ if\ x\ge 4 \end{cases} \tag{16}$$

$$\mu_{high}\left(x\right)=\begin{cases} 0,\ if\ x<2.5 \\ \dfrac{x-2.5}{2.5},\ if\ 2.5\le x\le 5 \\ 0,\ if\ x>5 \end{cases} \tag{17}$$

Using 4 Output Membership Functions: For 4 output membership functions we used equation 18, 19, 20, and 21.

$$\mu_{very\_low}\left(x\right)=\begin{cases} 0,\ if\ x<0 \\ \dfrac{1.5-x}{1.5},\ if\ 0\le x\le 1.5 \\ 0,\ if\ x\ge 1.5 \end{cases} \tag{18}$$

$$\mu_{low}\left(x\right)=\begin{cases} 0,\ if\ x\le 0.25 \\ \dfrac{x-0.25}{1.5},\ if\ 0.25\le x\le 1.75 \\ \dfrac{3.25-x}{1.5},\ if\ 1.75\le x\le 3.25 \\ 0,\ if\ x\ge 3.25 \end{cases} \tag{19}$$

$$\mu_{medium}\left(x\right)=\begin{cases} 0,\ if\ x\le 1.75 \\ \dfrac{x-1.75}{1.5},\ if\ 1.75\le x\le 3.25 \\ \dfrac{4.75-x}{1.5},\ if\ 3.25\le x\le 4.75 \\ 0,\ if\ x\ge 4.75 \end{cases} \tag{20}$$

$$\mu_{high}\left(x\right)=\begin{cases} 0,\ if\ x<3.5 \\ \dfrac{x-3.5}{1.5},\ if\ 3.5\le x\le 5 \\ 0,\ if\ x>5 \end{cases} \tag{21}$$

Using 5 Output Membership Functions: Equation 22, 23, 24, 25, and 26 were used as 5 membership functions of fuzzy output.

$$\mu_{very\_low}\left(x\right)=\begin{cases} 0,\ if\ x<0 \\ \dfrac{1.25-x}{1.25},\ if\ 0\le x\le 1.25 \\ 0,\ if\ x\ge 1.25 \end{cases} \tag{22}$$

$$\mu_{low}\left(x\right)=\begin{cases} 0,\ if\ x\le 0 \\ \dfrac{x}{1.25},\ if\ 0\le x\le 1.25 \\ \dfrac{2.5-x}{1.25},\ if\ 1.25\le x\le 2.5 \\ 0,\ if\ x\ge 2.5 \end{cases} \tag{23}$$

$$\mu_{medium}\left(x\right)=\begin{cases} 0,\ if\ x\le 1.25 \\ \dfrac{x-1.25}{1.5},\ if\ 1.25\le x\le 2.5 \\ \dfrac{3.75-x}{1.25},\ if\ 2.5\le x\le 3.75 \\ 0,\ if\ x\ge 3.75 \end{cases} \tag{24}$$

$$\mu_{high}\left(x\right)=\begin{cases} 0,\ if\ x\le 2.5 \\ \dfrac{x-2.5}{1.25},\ if\ 2.5\le x\le 3.75 \\ \dfrac{5-x}{1.25},\ if\ 3.75\le x\le 5 \\ 0,\ if\ x\ge 5 \end{cases} \tag{25}$$

$$\mu_{very\_high}\left(x\right)=\begin{cases} 0,\ if\ x<3.75 \\ \dfrac{x-3.75}{1.25},\ if\ 3.75\le x\le 5 \\ 0,\ if\ x>5 \end{cases} \tag{26}$$

4.2. Output Surfaces for Triangular Membership Functions

The output surfaces for 3, 4, and 5 triangular output membership functions are shown in figure 2, 3, and 4 respectively.

4.3. Gaussian Membership Functions

Inputs

Average Rating by User (avgRatingByUser): Equation 27, 28, and 29 are used as Gaussian membership functions of this input.

Figure 2. Output surface using 3 triangular output membership functions

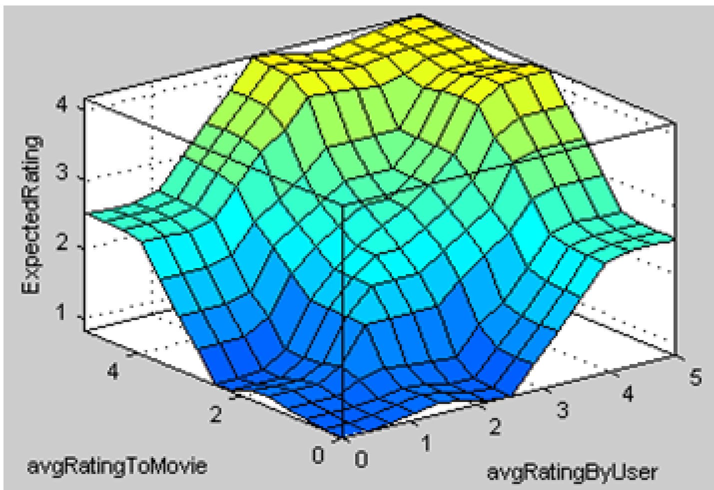


Figure 3. Output surface using 4 triangular output membership functions

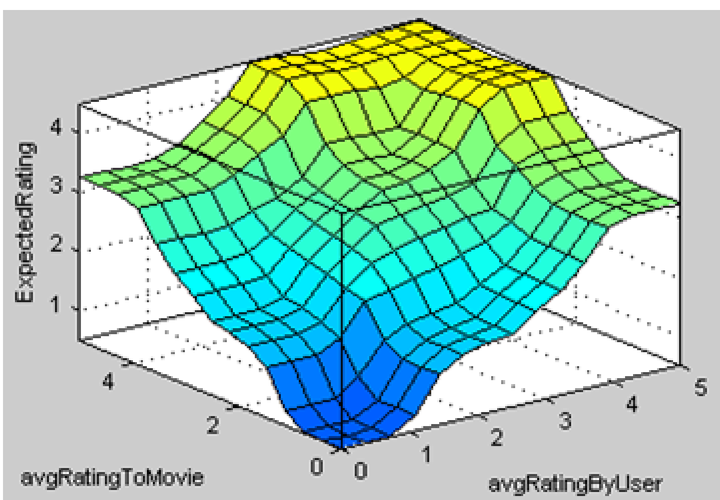
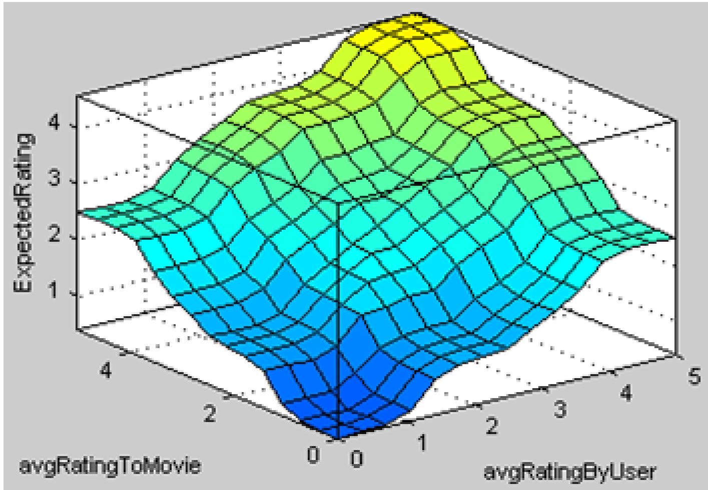


Figure 4. Output surface using 5 triangular output membership functions



$$\mu_{low}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x}{0.8493}\right)^2} \tag{27}$$

$$\mu_{medium}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-2.5}{0.48}\right)^2} \tag{28}$$

$$\mu_{high}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-5}{0.8493}\right)^2} \tag{29}$$

Average Rating to Movie (avgRatingToMovie): Equation 30, 31, and 32 are used as Gaussian membership functions of this input.

$$\mu_{low}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x}{0.8493}\right)^2} \tag{30}$$

$$\mu_{medium}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-2.5}{0.48}\right)^2} \tag{31}$$

$$\mu_{high}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-5}{0.8493}\right)^2} \tag{32}$$

Output

Using 3 Output Membership Functions: For 3 Gaussian output membership functions we used equation 33, 34, and 35.

$$\mu_{low}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x}{0.8493}\right)^2} \tag{33}$$

$$\mu_{medium}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-2.5}{0.48}\right)^2} \tag{34}$$

$$\mu_{high}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-5}{0.8493}\right)^2} \tag{35}$$

Using 4 Output Membership Functions: For 4 Gaussian output membership functions equation 36, 37, 38, and 39 are used.

$$\mu_{very\_low}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x}{0.48}\right)^2} \tag{36}$$

$$\mu_{low}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-1.75}{0.48}\right)^2} \tag{37}$$

$$\mu_{medium}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-3.25}{0.48}\right)^2} \tag{38}$$

$$\mu_{high}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-5}{0.48}\right)^2} \tag{39}$$

Using 5 Output Membership Functions: Equation 40, 41, 42, 43, and 44 are used as 5 Gaussian output membership functions.

$$\mu_{very\_low}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x}{0.42}\right)^2} \tag{40}$$

$$\mu_{low}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-1.25}{0.42}\right)^2} \tag{41}$$

$$\mu_{medium}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-2.5}{0.42}\right)^2} \tag{42}$$

$$\mu_{high}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-3.75}{0.42}\right)^2} \tag{43}$$

$$\mu_{very\_high}\left(x\right)=e^{-\frac{1}{2}\left(\frac{x-5}{0.42}\right)^2} \tag{44}$$

4.4. Output Surfaces for Gaussian Membership Functions

Figure 5, 6, and 7 show output surfaces for 3, 4, and 5 Gaussian output membership functions respectively.

4.5. Trapezoidal Membership Function

Inputs

Average Rating by User (avgRatingByUser): Equation 45, 46, and 47 are used as Trapezoidal membership functions of this input.

$$\mu_{low}\left(x\right)=\begin{cases} 0, \text{ } if \text{ } x<0 \\ 1, \text{ } if \text{ } 0\le x\le 1 \\ 2-x, \text{ } if \text{ } 1\le x\le 2 \\ 0, \text{ } if \text{ } x\ge 2 \end{cases} \tag{45}$$

$$\mu_{medium}\left(x\right)=\begin{cases} 0, \text{ } if \text{ } x\le 1 \\ x-1, \text{ } if \text{ } 1\le x\le 2 \\ 1, \text{ } if \text{ } 2\le x\le 3 \\ 4-x, \text{ } if \text{ } 3\le x\le 4 \\ 0, \text{ } if \text{ } x\ge 4 \end{cases} \tag{46}$$

$$\mu_{high}\left(x\right)=\begin{cases} 0, \text{ } if \text{ } x\le 3 \\ x-3, \text{ } if \text{ } 3\le x\le 4 \\ 1, \text{ } if \text{ } 4\le x\le 5 \\ 0, \text{ } if \text{ } x>5 \end{cases} \tag{47}$$

Average Rating to Movie (avgRatingToMovie): Equation 48, 49, and 50 are used as Trapezoidal membership functions for this input.

Figure 5. Output surface using 3 Gaussian output membership functions

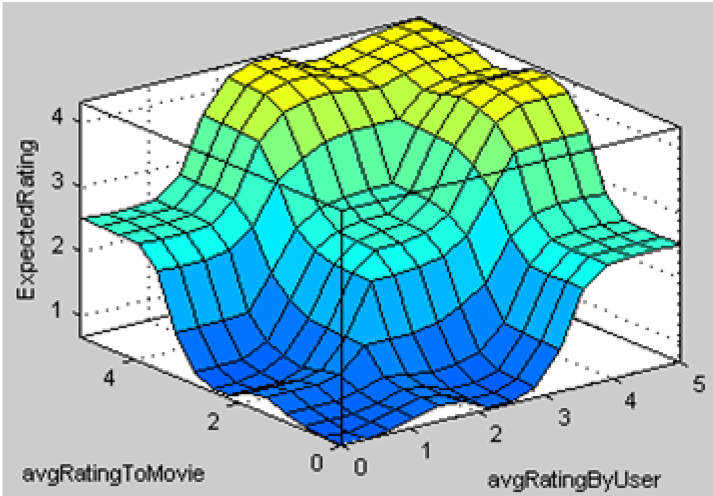


Figure 6. Output surface using 4 Gaussian output membership functions

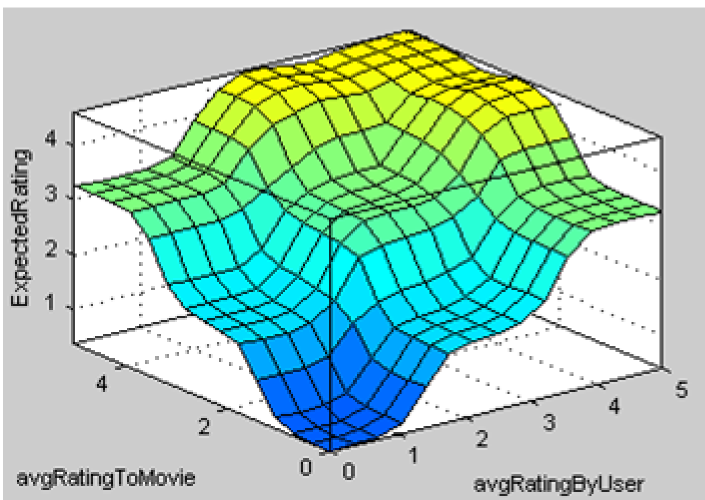
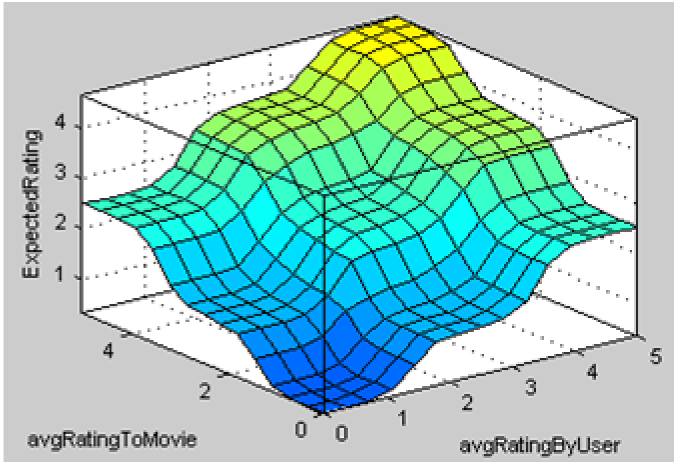


Figure 7. Output surface using 5 Gaussian output membership functions



$$\mu_{low}(x) = \begin{cases} 0, \ if \ x < 0 \\ 1, \ if \ 0 \le x \le 1 \\ 2 - x, \ if \ 1 \le x \le 2 \\ 0, \ if \ x \ge 2 \end{cases} \tag{48}$$

$$\mu_{medium}(x) = \begin{cases} 0, \ if \ x \le 1 \\ x - 1, \ if \ 1 \le x \le 2 \\ 1, \ if \ 2 \le x \le 3 \\ 4 - x, \ if \ 3 \le x \le 4 \\ 0, \ if \ x \ge 4 \end{cases} \tag{49}$$

$$\mu_{high}(x) = \begin{cases} 0, \ if \ x \le 3 \\ x - 3, \ if \ 3 \le x \le 4 \\ 1, \ if \ 4 \le x \le 5 \\ 0, \ if \ x > 5 \end{cases} \tag{50}$$

Output

Using 3 Output Membership Functions: Equation 51, 52, and 53 are used to describe Trapezoidal output membership functions when we used 3 output membership functions.

$$\mu_{low}\left(x\right)=\begin{cases} 0,\ if\ x<0 \\ 1,\ if\ 0\le x\le 1 \\ 2-x,\ if\ 1\le x\le 2 \\ 0,\ if\ x\ge 2 \end{cases} \tag{51}$$

$$\mu_{medium}\left(x\right)=\begin{cases} 0,\ if\ x\le 1 \\ x-1,\ if\ 1\le x\le 2 \\ 1,\ if\ 2\le x\le 3 \\ 4-x,\ if\ 3\le x\le 4 \\ 0,\ if\ x\ge 4 \end{cases} \tag{52}$$

$$\mu_{high}\left(x\right)=\begin{cases} 0,\ if\ x\le 3 \\ x-3,\ if\ 3\le x\le 4 \\ 1,\ if\ 4\le x\le 5 \\ 0,\ if\ x>5 \end{cases} \tag{53}$$

Using 4 Output Membership Functions: Equation 54, 55, 56, and 57 are used to describe Trapezoidal output membership functions when we used 4 output membership functions.

$$\mu_{very\_low}\left(x\right)=\begin{cases} 0,\ if\ x<0 \\ 1,\ if\ 0\le x\le 0.5 \\ 1.5-x,\ if\ 0.5\le x\le 1.5 \\ 0,\ if\ x\ge 1.5 \end{cases} \tag{54}$$

$$\mu_{low}\left(x\right)=\begin{cases} 0,\ if\ x\le 0.5 \\ x-0.5,\ if\ 0.5\le x\le 1.5 \\ 1,\ if\ 1.5\le x\le 2 \\ 3-x,\ if\ 2\le x\le 3 \\ 0,\ if\ x\ge 3 \end{cases} \tag{55}$$

$$\mu_{medium}\left(x\right)=\begin{cases} 0,\ if\ x\le 2 \\ x-2,\ if\ 2\le x\le 3 \\ 1,\ if\ 3\le x\le 3.5 \\ 4-x,\ if\ 3.5\le x\le 4.5 \\ 0,\ if\ x\ge 4.5 \end{cases} \tag{56}$$

$$\mu_{high}\left(x\right)=\begin{cases} 0,\ if\ x\le 3.5 \\ x-3.5,\ if\ 3.5\le x\le 4.5 \\ 1,\ if\ 4.5\le x\le 5 \\ 0,\ if\ x>5 \end{cases} \tag{57}$$

Using 5 Output Membership Functions: Equation 58, 59, 60, 61, and 62 are used to describe Trapezoidal output membership functions when we use 5 output membership functions.

$$\mu_{very\_low}\left(x\right)=\begin{cases} 0,\ if\ x<0 \\ 1,\ if\ 0\le x\le 0.75 \\ \dfrac{1.1-x}{0.35},\ if\ 0.75\le x\le 1.1 \\ 0,\ if\ x\ge 1.1 \end{cases} \tag{58}$$

$$\mu_{low}\left(x\right)=\begin{cases} 0,\ if\ x\le 0.75 \\ \dfrac{x-0.75}{0.35},\ if\ 0.75\le x\le 1.1 \\ 1,\ if\ 1.1\le x\le 1.85 \\ \dfrac{2.2-x}{0.35},\ if\ 1.85\le x\le 2.2 \\ 0,\ if\ x\ge 2.2 \end{cases} \tag{59}$$

$$\mu_{medium}\left(x\right)=\begin{cases} 0,\ if\ x\le 1.85 \\ \dfrac{x-1.85}{0.35},\ if\ 1.85\le x\le 2.2 \\ 1,\ if\ 2.2\le x\le 2.95 \\ \dfrac{3.3-x}{0.35},\ if\ 2.95\le x\le 3.3 \\ 0,\ if\ x\ge 3.3 \end{cases} \tag{60}$$

$$\mu_{high}\left(x\right)=\begin{cases} 0,\ if\ x\le 2.95 \\ \dfrac{x-2.95}{0.35},\ if\ 2.95\le x\le 3.3 \\ 1,\ if\ 3.3\le x\le 4.05 \\ \dfrac{4.4-x}{0.35},\ if\ 4.05\le x\le 4.4 \\ 0,\ if\ x\ge 4.4 \end{cases} \tag{61}$$

$$\mu_{very\_high}(x) = \begin{cases} 0, \ if\ x \le 4.05 \\ \dfrac{x-4.05}{0.35}, \ if\ 4.05 \le x \le 4.4 \\ 1, \ if\ 4.4 \le x \le 5 \\ 0, \ if\ x \ge 5 \end{cases} \quad (62)$$

4.6. Output Surfaces for Trapezoidal Membership Functions

Figure 8, 9, and 10 show the output surfaces for 3, 4, and 5 Trapezoidal output membership functions respectively.

5. ANFIS SYSTEM

ANFIS (Adaptive Neuro-Fuzzy Inference System) is a neural network based on the fuzzy inference system. It has several advantages over the Mamdani Method and it works well with optimization and adaptive techniques. It also has guaranteed continuity of the output surface. In ANFIS, the membership functions and parameters are tuned (adjusted) using either back-propagation algorithm only or with combining with a least squares type method. This adjustment allows the fuzzy system to learn from the data they model.

In our ANFIS system, we have used 3 types of membership functions. They are triangular, trapezoid and Gaussian functions. We have 2 inputs here similar to our fuzzy inference system and we used 3 membership functions for each. For our output, we have used linear type membership function. We have used 36 different combinations. Here for elaboration, we have given the initial and final membership functions where inputs were measured using Pearson Correlation Coefficient with K=5 with triangular membership function in equation 63 to 83. The ANFIS system generated 9 rules for inference system automatically. During training, we have used the hybrid optimization method.

Figure 8. Output surface using 3 trapezoidal output membership functions

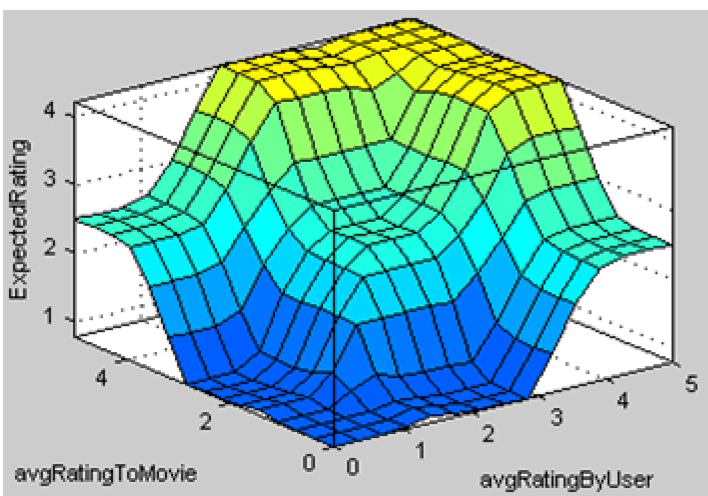


Figure 9. Output surface using 4 trapezoidal output membership functions

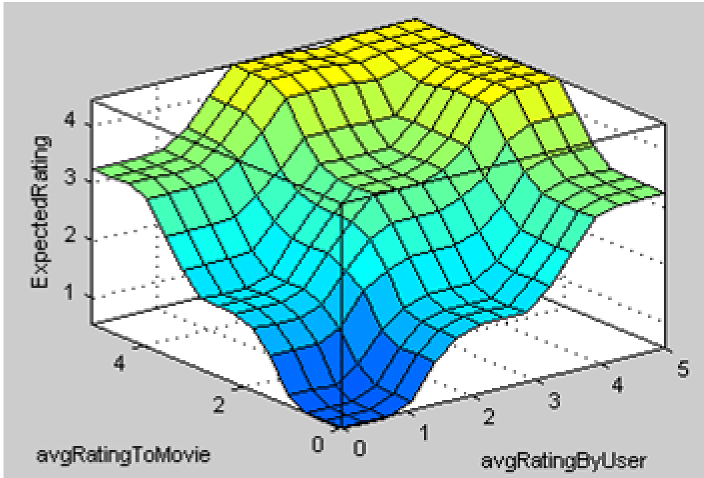
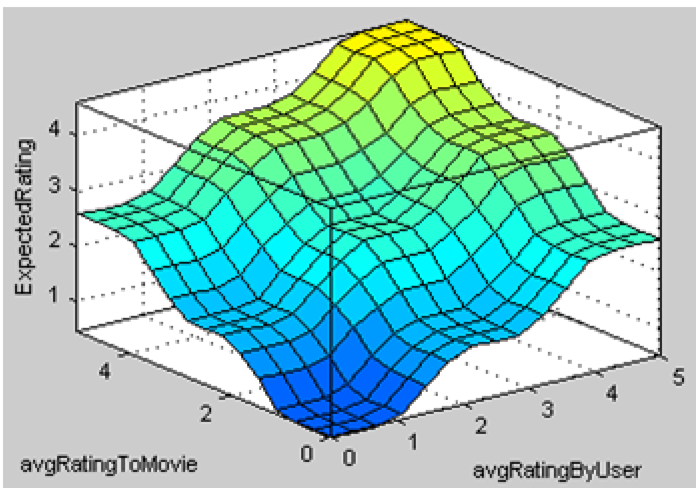


Figure 10. Output surface using 5 trapezoidal output membership functions



5.1. Initial Membership Functions

5.1.1. Average Rating by User (Input1)

$$\mu_{mf1}(x)=\begin{cases} 0,\ if\ x\le 0.2 \\ \dfrac{x-0.2}{1.6},\ if\ 0.2\le x\le 1.8 \\ \dfrac{3.417-x}{1.617},\ if\ 1.8\le x\le 3.417 \\ 0,\ if\ x\ge 3.417 \end{cases} \tag{63}$$

$$\mu_{mf2}(x)=\begin{cases} 0,\ if\ x\le 1.799 \\ \dfrac{x-1.799}{1.599},\ if\ 1.799\le x\le 3.398 \\ \dfrac{5.001-x}{1.603},\ if\ 3.398\le x\le 5.001 \\ 0,\ if\ x\ge 5.001 \end{cases} \tag{64}$$

$$\mu_{mf3}(x)=\begin{cases} 0,\ if\ x\le 3.326 \\ \dfrac{x-3.326}{1.67},\ if\ 3.326\le x\le 4.996 \\ \dfrac{6.6-x}{1.604},\ if\ 4.996\le x\le 6.6 \\ 0,\ if\ x\ge 6.6 \end{cases} \tag{65}$$

5.1.2. Average Rating to Movies (Input2)

$$\mu_{mf1}(x)=\begin{cases} 0,\ if\ x\le 0.5 \\ \dfrac{x-0.5}{1.5},\ if\ 0.5\le x\le 2 \\ \dfrac{3.5-x}{1.5},\ if\ 2\le x\le 3.5 \\ 0,\ if\ x\ge 3.5 \end{cases} \tag{66}$$

$$\mu_{mf2}(x)=\begin{cases} 0,\ if\ x\le 2 \\ \dfrac{x-2}{1.5},\ if\ 2\le x\le 3.5 \\ \dfrac{5-x}{1.5},\ if\ 3.5\le x\le 5 \\ 0,\ if\ x\ge 5 \end{cases} \tag{67}$$

$$\mu_{mf3}\left(x\right)=\begin{cases} 0,\ if\ x\le 3.5 \\ \dfrac{x-3.5}{1.5},\ if\ 3.5\le x\le 5 \\ \dfrac{6.5-x}{1.5},\ if\ 5\le x\le 6.5 \\ 0,\ if\ x\ge 6.5 \end{cases} \tag{68}$$

5.2. Final Membership Functions

5.2.1. Average Rating by User (Input1)

$$\mu_{mf1}\left(x\right)=\begin{cases} 0,\ if\ x\le 0.2 \\ \dfrac{x-0.2}{1.6},\ if\ 0.2\le x\le 1.8 \\ \dfrac{3.417-x}{1.617},\ if\ 1.8\le x\le 3.417 \\ 0,\ if\ x\ge 3.417 \end{cases} \tag{69}$$

$$\mu_{mf2}\left(x\right)=\begin{cases} 0,\ if\ x\le 1.799 \\ \dfrac{x-1.799}{1.599},\ if\ 1.799\le x\le 3.398 \\ \dfrac{5.001-x}{1.603},\ if\ 3.398\le x\le 5.001 \\ 0,\ if\ x\ge 5.001 \end{cases} \tag{70}$$

$$\mu_{mf3}\left(x\right)=\begin{cases} 0,\ if\ x\le 3.326 \\ \dfrac{x-3.326}{1.67},\ if\ 3.326\le x\le 4.996 \\ \dfrac{6.6-x}{1.604},\ if\ 4.996\le x\le 6.6 \\ 0,\ if\ x\ge 6.6 \end{cases} \tag{71}$$

5.2.2. Average Rating to Movies (Input2)

$$\mu_{mf1}\left(x\right)=\begin{cases} 0,\ if\ x\le 0.5 \\ \dfrac{x-0.5}{1.5001},\ if\ 0.5\le x\le 2.001 \\ \dfrac{3.492-x}{1.533},\ if\ 2.001\le x\le 3.534 \\ 0,\ if\ x\ge 3.534 \end{cases} \tag{72}$$

$$\mu_{mf2}(x)=\begin{cases} 0, \; if \; x \le 1.999 \\ \dfrac{x-1.999}{1.501}, \; if \; 1.999 \le x \le 3.5 \\ \dfrac{5.002-x}{1.502}, \; if \; 3.5 \le x \le 5.002 \\ 0, \; if \; x \ge 5.002 \end{cases} \quad (73)$$

$$\mu_{mf3}(x)=\begin{cases} 0, \; if \; x \le 3.49 \\ \dfrac{x-3.49}{1.509}, \; if \; 3.49 \le x \le 4.999 \\ \dfrac{6.5-x}{1.501}, \; if \; 4.999 \le x \le 6.5 \\ 0, \; if \; x \ge 6.5 \end{cases} \quad (74)$$

5.3. Output Membership Functions

$$\text{MF1} = -9.23*\text{INPUT1} + 38.23*\text{INPUT2} - 111.3 \quad (75)$$

$$\text{MF2} = 13.66*\text{INPUT1} - 15.95*\text{INPUT2} + 31.87 \quad (76)$$

$$\text{MF3} = 129.9*\text{INPUT1} - 27.29*\text{INPUT2} - 97.35 \quad (77)$$

$$\text{MF4} = -19.36*\text{INPUT1} - 38.21*\text{INPUT2} + 148.9 \quad (78)$$

$$\text{MF5} = 20.57*\text{INPUT1} - 34.1*\text{INPUT2} + 53.88 \quad (79)$$

$$\text{MF6} = 142.5*\text{INPUT1} - 29.78*\text{INPUT2} - 322.7 \quad (80)$$

$$\text{MF7} = -19.75*\text{INPUT1} + 29.85*\text{INPUT2} + 42.54 \quad (81)$$

$$\text{MF8} = 20.72*\text{INPUT1} + 27.56*\text{INPUT2} - 196.2 \quad (82)$$

$$\text{MF9} = 144*\text{INPUT1} + 23.41*\text{INPUT2} - 834.6 \quad (83)$$

6. DATASET

For the purpose of this research, we have extracted data provided by the organization MovieLens which is a concern of GroupLens (GroupLens, n.d.). GroupLens is a research lab in the Department of Computer Science and Engineering at the University of Minnesota, specializing in recommender systems, online communities, mobile and ubiquitous technologies, digital libraries, and local geographic information systems. There are many faculty members, graduate students and staffs working behind the development of this program.

MovieLens provide movie recommender service where a user can register for an account and rate their favorite movies. When a new user logs in, they are given a series of movies to rate on a scale of 1 to 5. The system compares the ratings to those of other users with similar tests and then recommends them movies according to their taste.

There were several datasets which they provide to public for research purposes. The datasets were collected by MovieLens over various periods of time. Dataset volume ranges from 100,000 ratings to 10 million ratings. We have used the dataset that has 1 million rating data from 6000 users and 4000 movies. The dataset is available at Grouplens Datasets (GroupLens, 2013).

The dataset had 3 files:

6.1. Ratings File

- *ratings.dat* – contains the rating (on scale 5) information in following format:

  UserID:: MovieID:: Rating:: Timestamp

UserID is the unique ID of an user and MovieID is the unique ID of a movie. Rating is the value that the user has given to that particular movie. Timestamp is the time and date when the user placed the rating.

6.2. Movies File

- *movies.dat* – this file contains information about the movies. The format is:

  MovieID:: Title:: Genres

Here, MovieID is the unique ID of the movie and title is the name of the movie. Genres field contained several genres that the movies belong to.

The Genres are selected from the Genre list presented in Box 1.

Box 1. Genre list

Action	Film-Noir
Adventure	Horror
Animation	Musical
Children's	Mystery
Comedy	Romance
Crime	Sci-Fi
Documentary	Thriller
Drama	War
Fantasy	Western

6.2. Users File

users.dat – contains the users' information in the following format:
UserID:: Gender:: Age:: Occupation:: Zip-code

- Gender is denoted by a "M" for male and "F" for female
- Age is chosen from the following ranges:
  - 1: "Under 18"
  - 18: "18-24"
  - 25: "25-34"
  - 35: "35-44"
  - 45: "45-49"
  - 50: "50-55"
  - 56: "56+"
- Occupation is chosen from the following choices:
  - 0: "other" or not specified
  - 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"

Using these data, we have calculated the input parameters of our fuzzy inference system.

7. RESULTS

7.1. Using Fuzzy Inference System (FIS)

We have varied the input parameters to determine which combination gives the best performance. The value of K is selected among 5, 10, and 20 to choose the most similar neighbors. This is because with increasing number similar users in consideration, the output error may increase or decrease depending on the dataset. We certainly do not know where the dissimilarity begins.

We also used different input membership functions in hope to achieve better performance. To determine which distribution gives better performance, we have tried triangular, trapezoidal and Gaussian fuzzy membership functions to determine the distribution of the dataset. For input to our fuzzy inference system, we have used the ratings of the user 314 from our dataset who has rated 293 movies.

Output membership functions (mf) are varied using different combinations. In the result tables, mf # 3 represents {LOW, MEDIUM, HIGH}, mf # 4 represents {VERY_LOW, LOW, MEDIUM, HIGH} and mf # 5 means {VERY_LOW, LOW, MEDIUM, HIGH, VERY_HIGH}. According to overall results, when less number of fuzzy value in membership functions are used, the greater the error. As we increased the fuzzy values in membership functions, the error decreased. For example mf # 5 gives better results compared to mf # 3. We have calculated the output error using all possible combinations and the results analysis are as below:

7.1.1. Using Manhattan Distance

From Table 14, 15, and 16 we see, using Manhattan Distance for calculating input parameters, the best results are achieved when using K=5, which means the expected rating of the target user is measured taking in account the choices of the top 5 most similar users and movies. Performance increases when 3 fuzzy value output function is used using the trapezoidal membership function.

7.1.2. Using Euclidean Distance

From Table 17, 18, and 19 we see that the best results are found when K is set to 5 and 3 fuzzy value output function is used. The trapezoidal distribution gives the most correct prediction of expected rating in this case. Here the error is increasing with increasing value of K. This means the dissimilarity increases when more users are considered.

7.1.3. Using Pearson Correlation

From Table 20, 21, and 22 we see that Pearson Correlation provides the best results. The results are consistent. It can be seen that with increasing value of K, the error does not increase much comparing to Euclidean Distance method. In this case, the best performance is achieved when the Trapezoidal distribution is used with 3 fuzzy values in output functions and K is set to 10.

7.1.4. Using Cosine Similarity

When Cosine Similarity is used for calculating input parameters, from Table 23, 24, and 25 we see that the best performance is achieved when K is set to 20 and 3 fuzzy values is used in output fuzzy function. The Trapezoidal distribution gives the best results in this case. Among all the techniques to determine input parameters, Cosine Similarity gives the worst performance.

Analyzing these data, it can be concluded that Pearson Correlation method gives the most accurate results. Therefore it is the best method to recommend users. A comparison of all the values of Table 14 to 25 is given in Figure 11.

Now we know the expected rating of a movie from the defuzzified crisp value of Fuzzy Inference System output. This expected rating is then used to sort a list of movies in descending order to get the highly recommended movies.

Table 14. RMSE Error for Different Membership Functions with K=5 and Manhattan Distance using FIS

mf type	mf #3	mf #4	mf #5
triangular	0.889121	0.948088	0.901725
trapezoidal	0.880194	0.957016	0.914499
gaussian	0.892261	0.991069	0.914835

Table 15. RMSE Error for Different Membership Functions with K=10 and Manhattan Distance using FIS

mf type	mf #3	mf #4	mf #5
triangular	0.897824	0.959737	0.914979
trapezoidal	0.889759	0.971898	0.934191
gaussian	0.917097	1.015669	0.940839

Table 16. RMSE Error for Different Membership Functions with K=20 and Manhattan Distance using FIS

mf type	mf #3	mf #4	mf #5
triangular	0.918819	0.969982	0.936395
trapezoidal	0.904923	0.987918	0.954596
gaussian	0.953892	1.049411	0.976026

Table 17. RMSE Error for Different Membership Functions with K=5 and Euclidean Distance using FIS

mf type	mf #3	mf #4	mf #5
triangular	0.865487	0.926347	0.88818
trapezoidal	0.854285	0.940147	0.901139
gaussian	0.874062	0.976619	0.893311

Table 18. RMSE Error for Different Membership Functions with K=10 Euclidean Distance using FIS

mf type	mf #3	mf #4	mf #5
triangular	0.870109	0.940856	0.894391
trapezoidal	0.855913	0.951108	0.911793
gaussian	0.882481	0.989928	0.914016

Table 19. RMSE Error for Different Membership Functions with K=20 Euclidean Distance using FIS

mf type	mf #3	mf #4	mf #5
triangular	0.899494	0.95134	0.91164
trapezoidal	0.878421	0.96887	0.925297
gaussian	0.94105	1.037425	0.947824

Table 20. RMSE Error for Different Membership Functions with K=5 and Pearson Correlation using FIS

mf type	mf #3	mf #4	mf #5
triangular	0.825977	0.909432	0.853109
trapezoidal	0.804346	0.915076	0.877785
gaussian	0.825365	0.942536	0.883128

Table 21. RMSE Error for Different Membership Functions with K=105 and Pearson Correlation using FIS

mf type	mf #3	mf #4	mf #5
triangular	0.799763	0.878413	0.838811
trapezoidal	0.765881	0.888888	0.852166
gaussian	0.807671	0.929772	0.852695

Table 22. RMSE Error for Different Membership Functions with K=205 and Pearson Correlation using FIS

mf type	mf #3	mf #4	mf #5
triangular	0.870288	0.914263	0.869483
trapezoidal	0.841135	0.93675	0.86749
gaussian	0.926002	1.012041	0.901284

Table 23. RMSE Error for Different Membership Functions with K=5 and Cosine Similarity using FIS

mf type	mf #3	mf #4	mf #5
triangular	0.99572	1.051525	0.970587
trapezoidal	1.016058	1.06022	1.002501
gaussian	1.013556	1.096192	1.003451

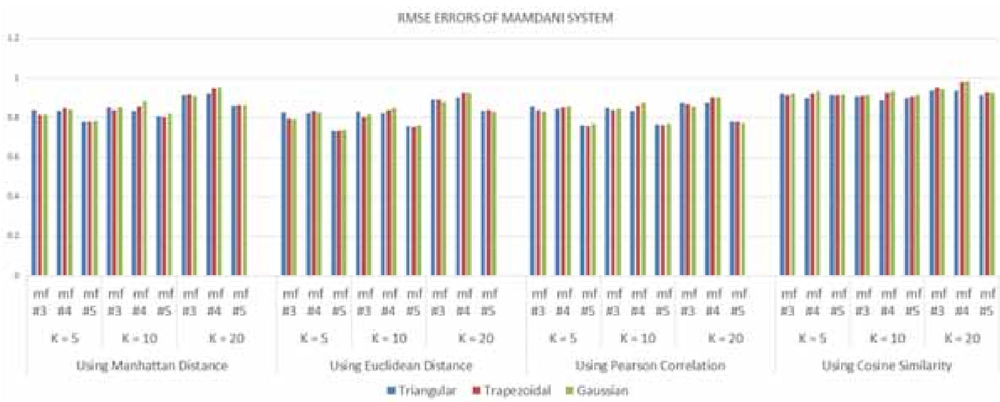
Table 24. RMSE Error for Different Membership Functions with K=10 and Cosine Similarity using FIS

mf type	mf #3	mf #4	mf #5
triangular	1.006888	1.049955	0.993354
trapezoidal	1.028912	1.060811	1.013779
gaussian	1.027374	1.103601	1.010767

Table 25. RMSE Error for Different Membership Functions with K=20 and Cosine Similarity using FIS

mf type	mf #3	mf #4	mf #5
triangular	0.919414	0.967741	0.967741
trapezoidal	0.905994	0.985372	0.956014
gaussian	0.954068	1.046761	0.978785

Figure 11. Comparison of Mamdani FIS system based on RMSE error



7.2. Using ANFIS

We have used the user from our dataset with ID 314 who has rated 293 movies. Our ANFIS system is trained with the data of this user. The training data has 2 input and 1 output. The first input was *AVERAGE RATING BY USER* and the second input was *AVERAGE RATING TO MOVIE*. We have calculated these input parameters using the same technique as we used earlier in the Mamdani Inference System. (e.g. K=5, K=10 and K=20 for each distance metrices).

The output column was original rating which the user 314 has given to the movies. Then we used 40 epochs to train our system with these datasets. For training, we have used 263 ratings among the 293 ratings that user with ID 314 has given.

For checking dataset, we have used the remaining 30 ratings from the user with ID 314 from our dataset. Similar to training dataset, the checking dataset also has 2 inputs and 1 output. The inputs are *AVERAGE RATING BY USER* and *AVERAGE RATING TO MOVIE* respectively. We have also calculated this like the training dataset. The output is the original rating of the user 314 to the 293 movies he rated.

The RMSE Error for different types of membership functions and different combinations of inputs are given from Table 26 to 37.

7.2.1. Using Manhattan Distance

Table 26. RMSE Error for Different Membership Functions with K=5 and Manhattan Distance using ANFIS

mf type	Training Error	Checking Error
triangular	0.8412	1.4575
trapezoidal	0.8434	0.9178
gaussian	0.8329	0.9737

Table 27. RMSE Error for Different Membership Functions with K=10 5 and Manhattan Distance using ANFIS

mf type	Training Error	Checking Error
triangular	0.8953	1.0026
trapezoidal	0.8973	0.9678
gaussian	0.8902	2.688

Table 28. RMSE Error for Different Membership Functions with K=205 and Manhattan Distance using ANFIS

mf type	Training Error	Checking Error
triangular	0.8701	1.048
trapezoidal	0.9089	1.103
gaussian	0.8892	0.9029

7.2.2. Using Euclidean Distance

Table 29. RMSE Error for Different Membership Functions with K=5 and Euclidean Distance using ANFIS

mf type	Training Error	Checking Error
triangular	0.7624	0.7947
trapezoidal	0.7824	0.7817
gaussian	0.7751	0.75

Table 30. RMSE Error for Different Membership Functions with K=10 and Euclidean Distance using ANFIS

mf type	Training Error	Checking Error
triangular	0.7997	0.7838
trapezoidal	0.8036	0.8226
gaussian	0.7934	0.9221

Table 31. RMSE Error for Different Membership Functions with K=20 and Euclidean Distance using ANFIS

mf type	Training Error	Checking Error
triangular	0.7935	1.769
trapezoidal	0.7991	2.4559
gaussian	0.8149	2.5019

7.2.3. Using Pearson Correlation Coefficient

Table 32.RMSE Error for Different Membership Functions with K=5 and Pearson Correlation using ANFIS

mf type	Training Error	Checking Error
triangular	0.7028	1.095
trapezoidal	0.7201	0.9696
gaussian	0.7202	1.2068

Table 33. RMSE Error for Different Membership Functions with K=10 and Pearson Correlation using ANFIS

mf type	Training Error	Checking Error
triangular	0.7014	0.7413
trapezoidal	0.697	0.749
gaussian	0.6973	0.7572

Table 34. RMSE Error for Different Membership Functions with K=20 and Pearson Correlation using ANFIS

mf type	Training Error	Checking Error
triangular	0.6759	0.7458
trapezoidal	0.6732	0.8048
gaussian	0.6795	0.9318

7.2.4. Using Cosine Similarity

Looking at these results, it can be seen that Pearson Correlation Coefficient gives the best results with the least training error and checking error. The best performance is achieved when selecting 10 neighbors and input membership functions are trapezoidal. Figure 12 shows a comparison of all the values of Table 26 to 37.

Table 35. RMSE Error for Different Membership Functions with K=5 and Cosine Similarity using ANFIS

mf type	Training Error	Checking Error
triangular	1.0213	1.0709
trapezoidal	1.001	1.1937
gaussian	1.0092	1.1698

Table 36. RMSE Error for Different Membership Functions with K=10 and Cosine Similarity using ANFIS

mf type	Training Error	Checking Error
triangular	0.9641	1.0889
trapezoidal	0.9717	1.3508
gaussian	0.9701	1.1158

Table 37. RMSE Error for Different Membership Functions with K=20 and Cosine Similarity using ANFIS

mf type	Training Error	Checking Error
triangular	0.8701	1.048
trapezoidal	0.9089	1.103
gaussian	0.8892	0.9029

Figure 12. Comparison of ANFIS system based on RMSE Error

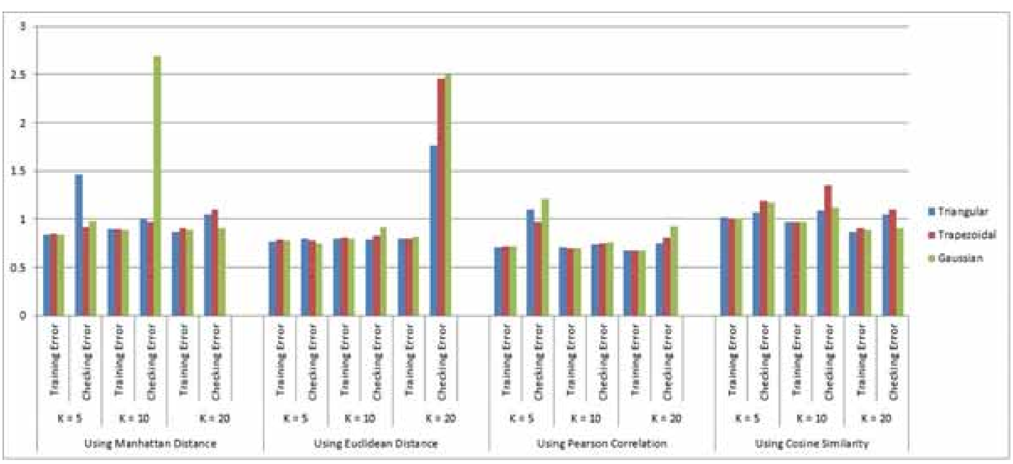
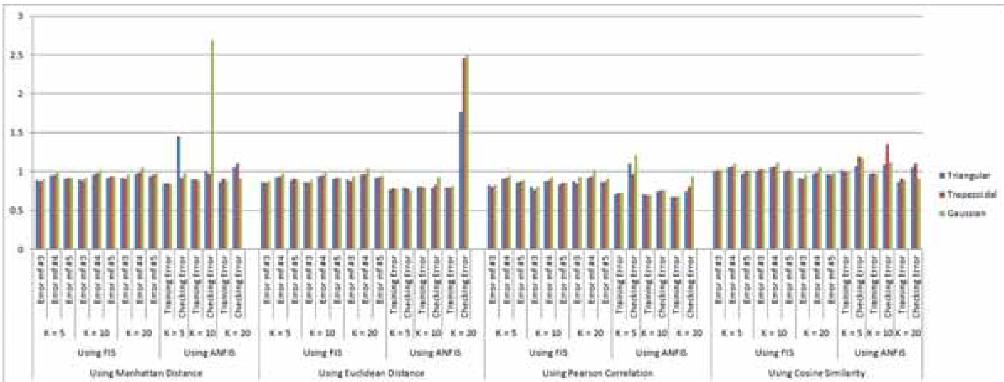


Figure 13. Comparison of FIS and ANFIS system based on RMSE error



8. CONCLUSION

Recommender systems are a very important tool to gain additional value in business. These systems help the user to find their desired items and alongside benefits the business. They help business generating more sales. The recommender systems are vastly improvised by the immense amount of user data existing in large organizations and will be further extended with the increasing number of users. Better technologies are needed that will improve the scalability of recommender systems.

Comparing between both Mamdani System and ANFIS System, we observed that ANFIS System with Pearson Correlation where K=10 gives the best performance.

In this paper, we experimented on the improvisation of recommender system using fuzzy logic. Our results shed light on the many possibilities that can be scored by implying fuzzy logic in recommendation.

REFERENCES

Breese, J., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence* (pp. 43--52).

Goldberg, D., Nichols, D., Oki, B., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, *35*(12), 61–70. doi:10.1145/138859.138867

GroupLens. (2013). *MovieLens*. Retrieved 9 September 2014, from http://www.grouplens.org/datasets/movielens

GroupLens. *GroupLens*. Retrieved 9 September 2014, from http://www.grouplens.org

He, J., & Chu, W. (2010). A social network-based recommender system (SNRS). *Annals of Information Systems*, *12*, 47–74.

Hill, W., Stead, L., Rosenstein, M., & Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 194--201). doi:10.1145/223904.223929

Resnick, P., & Varian, H. (1997). Recommender systems. *Communications of the ACM*, *40*(3), 56–58. doi:10.1145/245108.245121

Roy, D., & Kundu, A. (2013). Design of Movie Recommendation System by Means of Collaborative Filtering. *International Journal Of Emerging Technology And Advanced Engineering*, *3*(4), 67–72.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms, *World Wide Web Conference*, ACM, (pp. 285--295).

Semwal, V. S., Chakraborty, P., & Nandi, G. C. (2015). Less computationally intensive fuzzy logic (type-1)-based controller for humanoid push recovery. *Robotics and Autonomous Systems*, *63*(1), 122–135. doi:10.1016/j.robot.2014.09.001

Shardanand, U., & Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth", *Proceedings of 1995 Conference on Human Factors in Computing Systems (CHI)* (pp. 210--217). doi:10.1145/223904.223931