1. Overview

The paper proposes a framework to recommend users a large number of situations in rich websites. Table 2 indicates a general view of the proposed framework.

User situation	**Method**
Unauthorized users (homepage)	Top highest ratings
	Top views in a period
Unauthorized users (detailed item page)	Association rules
	Item of association
New authorized users	Demographic filtering
Frequent authorized users	Collaborative filtering and demographic filtering combination

Table 1 Methods used in each situation

At the first sight, users will see the home page of the system. There are two behaviors that they will continue: viewing the detailed page, logging or registering into the system. Without authentication, the system will suggest the same recommendation for everyone. In contrast, authenticated users will be supported.

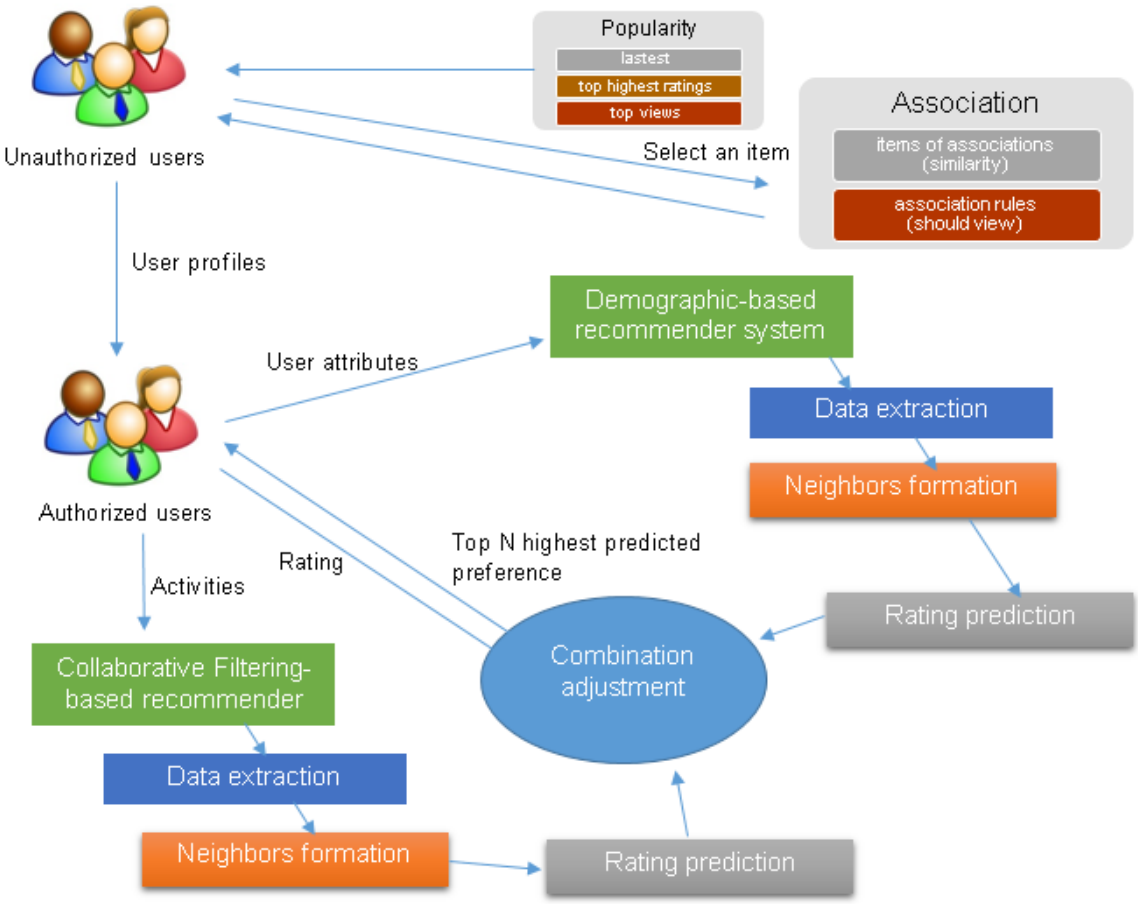

Figure 1 Overview of the proposed framework

Figure 7 reflects the overview of proposed framework for recommender systems. The system is improving while users are surfing on the web. When the users visit a website, they play as many roles in the website. For first time visitors, in the role of guests; they can experience the contents of the website. Then, the website will recommend some items which have the highest rating or visit mostly by using popularity–based method. This method runs by using simple statistics found on popular items in many perspectives.

Another method is demographic based method. Its data is based on the personal information of member such as: gender, age, occupation, salary, etc. As a result, when a guest registers to become a member, his personal information will be automatically updated for the database of website, and will be using as a key word to

sort other members who have the similar characteristics. Each of them has a collection of product they had seen and rated. This information will be sent to the recommender system to analyze and result the items which the new member will may interested in.

Continuously, the users experienced on the website, they saw and rated the items. It supplies valuable information for the work of collaborative filtering. In virtue of the rating of users, a group of users who has the same assessment will be created and named the similar group of users. Each similar user contains the list of items that are viewed and rated. In addition, a collection of similar users enhanced these lists and they will help z-score [6] to be improved. Next, each user will be evaluated carefully to personalize recommendation. A list of recommended items which was sorted by descending order of predicted rating will be introduced to users. Thus, they can easily view the recommended items and rate them. As a result, the website has the real rating of the users of that item. Subsequently, from predicted rating based on demographic and collaborative filtering as well as the real rating of the users in one item, perceptron method are utilized to figure out what that users prefer. While a large amount of users depend much on their demographic data; others depend on their collaborative activities on the website. According to the result, the website should combine flexibly two methods as well as research extensively each method for recommending the most accurate suggestions.

2. Details of the proposed framework

Firstly, unauthorized users will be treated by popularity method. The recommender system does statistics to calculate which items have the highest rating and which items have the most number of views (i.e. the most popular). Thus, all of unauthorized users have the same recommended items. Recommended items will be

altered efficiently and reflect the trend on the website which guests might like by filtering in a specific period. Similarly, the number of ratings of an item must be considered in a specific time and higher enough as a certain large number. To illustrate this issue, a new item, which was posted to website and it received the highest rating (5 stars) from only one user, will appear in the recommended items (items have the highest rating). It does not reflect the popularity. This framework proposes recommended items that were calculated in a period and were caught the threshold of views.

This formula shows average rating of an item. It is applied for all items. For each item *i*:

$$average\ rating\ (i) = (\sum_{k=1}^{n} r(u_k, i)\ )/\ n\ \forall\ n > c$$

where user u_k *(k=1-n), n*: total rating of item *i*. Each item *i* has n as a certain large number *c*, the value of n is determined by administrator of recommender systems. If *n* is less than that certain large number *c*, it should not be in recommended list for users. In this case, this administrator of these recommender systems have to define a period time that average rating (i) is calculated. This task reflects the popularity and the top growing items.

This table 3 shows the relationship between item, user, rating and time:

Table 2 User - Item rating matrix

Item	**User**	**Rating**	**Time**
i_1	u_1	4	2014/8/15 15:30:26
i_1	u_2	5	2014/8/16

			15:30:26
…	…	…	…

This SQL Statement will calculate the list for recommended items

select top N item

from

(

select item, avg(rating) as avg_rating

from tbl_rating

where time >= current_time - period_time and time <= current_time

)

where (select count(*) from tbl_rating t where item = t.item) >= threshold

order by avg_rating desc

The values that must be defined, including:

- Top N item (for example 10 for top 10 highest rating items)
- Period time (for example 1 month)
- Threshold: a certain large number (for example: 100 ratings per month, this number depends on total views of the whole system). This number should be high if the system traffic is large.

Similarity, top views in a period is a part of popularity recommended items. They are not highest rating, but they are paying attention. Many people see in these items.

Secondly, when unauthorized users are surfing the website, it creates a session that saves items that each user saw recently. At this time, association based method creates recommended items for that user.

The rating table 4 is convert into this view:

User	**List of item**
u_1	$i_1, i_5, i_8, \ldots i_k$
u_2	$i_2, i_9, i_{15}, i_{17} \ldots i_k$
…	…

Table 3 The point of view based on user's behaviors

The Apriori algorithm is a popular method that is used for mining frequent itemsets [14].

Let $I = \{I_1, I_2, I_3, \ldots\ldots, I_m\}$ be a set of items [14].

Let D be a set of transaction in a database where each transaction T is a set of items such that $T \subseteq I$ [14].

The form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, and $A \cap B = \emptyset$ [14]. The rule $A \Rightarrow B$ holds in the set of database transactions D with support s, where s is the percentage of transactions in D that contain $A \cup B$ which means the probability $P(A \cup B)$ indicates that a transaction contains the union of set A and set B [14]. In addition, the confidence c of the rule $A \Rightarrow B$ in the transaction set D is the percentage of transaction in D that containing A that also containing B too which means the conditional probability $P(B \mid A)$ [14]. Therefore, the rules that satisfy both a minimum support threshold and a minimum confidence threshold are called strong association rules [14]. The confidence c of rule $A \Rightarrow B$ can be derived from the

support count (the number of transactions that contain the itemset) of A and $A \cup B$ as is shown by the following equation [14]:

$$confidence\ (\mathrm{A} \Rightarrow \mathrm{B}) = P\ (B|A) = \frac{support\ (A\ \cup B)}{support\ (A)}$$

$$= \frac{support - count\ (A\ \cup B)}{support - count\ (A)}$$

Association rules generate a list of rules; for instance A⇒B (A and B are list of items). In addition, a session of a user contains a list of items that the user saw recently, it is called X. Association rules are filtered to map the condition: A is the list of items called X. Therefore, when unauthorized users view a detailed page of an item, a list of recommended items in B is showed. This reflects that people saw items in A should see items in B.

Moreover, in the detailed page, the website uses Pearson correlations to calculate the distance between two items. For each user, it sorts in descending order the Pearson value of other items to generate the most similar items to this item. These items are also used for recommending in detailed pages. It means that personalized recommendation for a Web session.

Thirdly, when users register, they provide personal information. Demographic filtering is initialized to find neighbors based on the similarity of personal information. Users also rate on website; thus, collaborative filtering is performed to find neighbors based on Pearson correlation:

$$sim(u_a, u_b) = \frac{\sum_i^m (p_{ai} - \bar{p}_a)(p_{bi} - \bar{p}_b)}{\sqrt{\sum_i^m (p_{ai} - \bar{p}_a)^2}\ \sqrt{\sum_i^m (p_{bi} - \bar{p}_b)^2}}$$

where p_{ai} is rating of u_a on item i, $\overline{p_a}$ is the average rating of user u_a, while m is number of items.

To predict the rating for each method, Z-score average [6] is used, p_{ai} is predicted the rating of user u_a on the item i:

$$p_{ai} = \bar{p}_a + \frac{\sum_{u=1}^{k} \frac{(p_{ui} - \bar{p}_u)}{\sigma_u} sim(u_a, u_u)}{\sum_{u=1}^{k} sim(u_a, u_u)}$$

The framework has a predicted rating for each method. For each user, perceptron neural network is performed to combine two of predicted ratings into a predicted rating. Perceptron neural network is a method that adjusts the distance of predited ratings in the past to the real ratings. In the future, predicted ratings will progress the same direction with users' preferences. This combination helps predicted ratings come to real rating nearly, thus the accuracy of the whole systems will be improved.

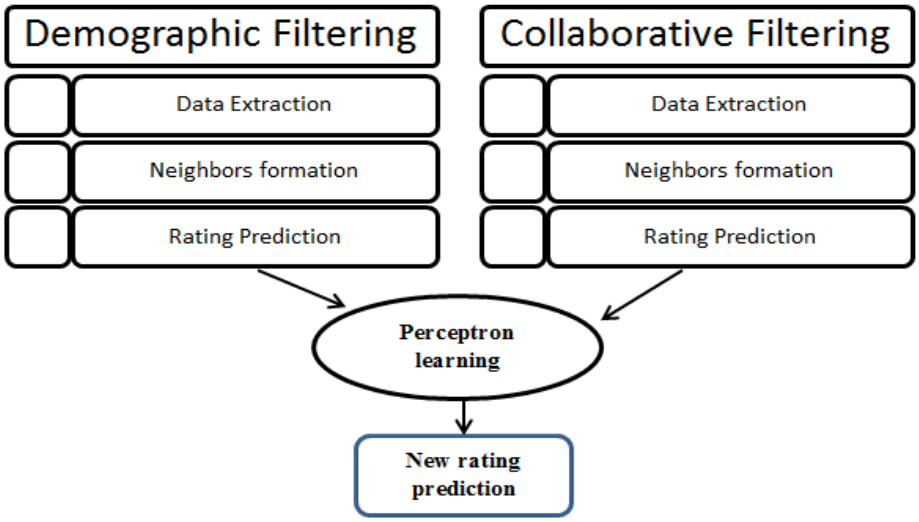

Figure 2 Combination of demographic filtering and collaborative filtering

Figure 8 shows the process of combination between demographic filtering and collaborative filtering using perceptron learning neural network and the perceptron neural network learning process follows these steps:

- Combining historical data and defining weight rate w_i

$$f(\sum_{i=1}^{j} w_i x_i)$$

- Updating weight rate for each historical data record
- Re-calcuating the result by applying new w_i, and repeat these steps with new data records.

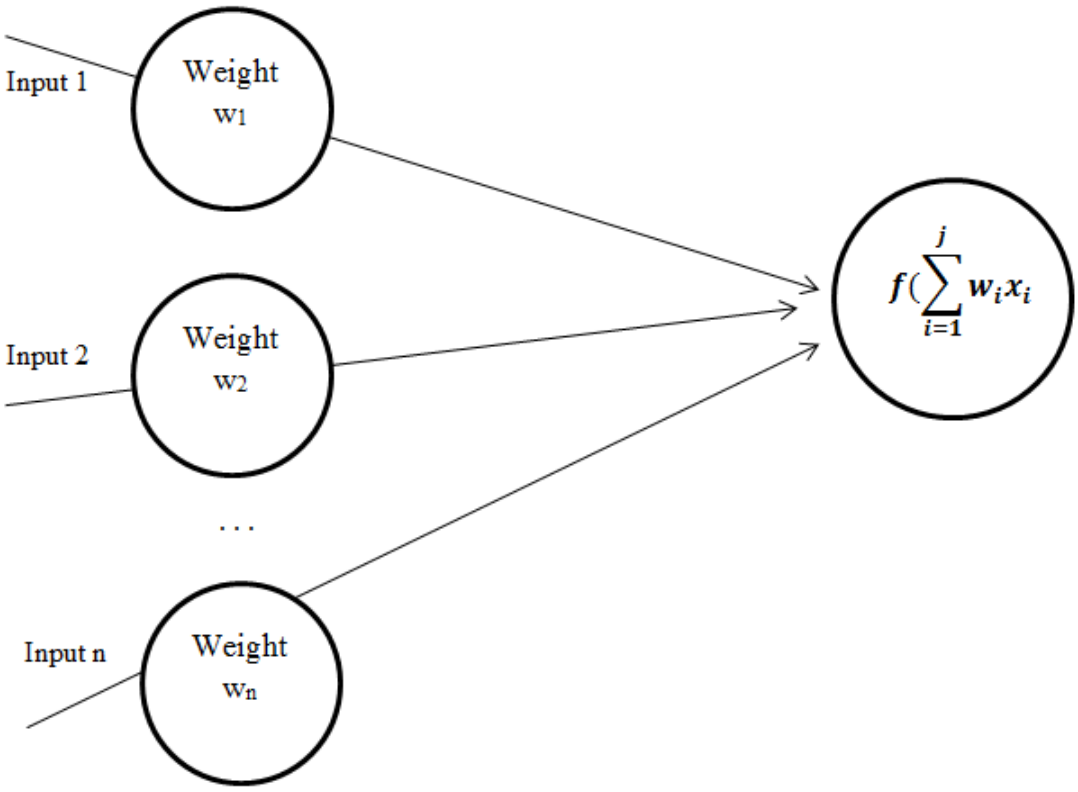

Figure 3 Perceptron learning process

To combine, using the process in figure 9 with 2 input values, randomly set weight *w1*, *w2* for each predicted rating; thus combined rating prediction is $n = w_1.p_1+w_2.p_2$. For each new rating of an active user, calculate new weights:

$$w_{1(new)} = w_1 + l\, p_1(p_{new} - n)$$

$$w_{2(new)} = w_2 + l\, p_2(p_{new} - n)$$

where l is learning rate, we suggest that l= *1/number of total rating,* p_i is predicted rating of method that maps to w_i, and p_{new} shows new actual rating. This formula is run immediately many times. The recommender system is improved after a new actual rating.

To illustrate the learning process, table 5 shows sample data for learning:

Learning rate = 1/300,000

No	CF	DE	REAL	W1	W2	ADJ
1	3.521328	4.276461	5	0.965447083	0.828925	6.944521
2	2.11433	2.805842	5	0.965424259	0.828897	4.366981
3	1.630384	1.618296	4	0.96542872	0.828903	2.91543
4	0.913157	2.084553	3	0.965434614	0.828909	2.609498
5	1.992577	2.825578	5	0.965435803	0.828912	4.26586

Table 4 Perceptron learning example

"No" field: denotes row number

"CF" : collaborative filtering prediction

"DE" : demographic filtering prediction

"REAL" : actual rating

"W1" : weight of collaborative filtering

"W2" : weight of demographic filtering

"ADJ" : new prediction by combination

W1 and W2 are randomly set at row number 1, *ADJ at row number 1 = CF x W1 + DE x W2*

At row number 2: let re-calculate:

$$W1_{row\ number\ 2} = W1_{row\ number\ 1} + learning\ rate \times CF_{row\ number\ 1} \times (REAL_{row\ number\ 1} - ADJ_{row\ number\ 1})$$

$$W2_{row\ number\ 2} = W2_{row\ number\ 1} + learning\ rate \times DE_{row\ number\ 1} \times (REAL_{row\ number\ 1} - ADJ_{row\ number\ 1})$$

$ADJ_{row\ number\ 2} = CF\ x\ W1 + DE\ x\ W2$ *(at the row number 2)*

Similarity, let calculate for rumber 3, 4, etc. This learning process is implemented for many times when w1 and w2 are not changed more than 0.00001 or specific small number (can be accepted).

After predicted ratings of combination are generated, they are used in rank order to suggest items for each user.

3. Experiment

a. Hardware and software

The machine used in this experiment:

Operating System: Windows 8.1 Pro 64-bit (6.3, Build 9600) (9600.winblue_gdr.131030-1505)

Language: English (Regional Setting: English)

System Manufacturer: Gigabyte Technology Co., Ltd.

System Model: To be filled by O.E.M.

BIOS: BIOS Date: 08/30/13 09:45:07 Ver: 04.06.05

Processor: Intel(R) Core(TM) i3-3220 CPU @ 3.30GHz (4 CPUs), ~3.3GHz

Memory: 4096MB RAM

Available OS Memory: 3990MB RAM

Page File: 4397MB used, 1705MB available

In order to implement the application for the demonstration, Visual Studio is used to program the application. Figure 10 show the interface of Visual Studio 2013

Figure 4 Visual Studio 2013 IDE for the experiment

Integrated development environment: Visual Studio 2013

Environment: Web Form

Framework: .NET Framework 4.0

Language: C#

Database: Microsoft SQL Server 2008

b. Dataset

The data is collected from MovieLens (http://grouplens.org/datasets/movielens/) for illustrating the proposal.

The original file contains 3,900 movies, 6,040 MovieLens, and 1,000,000 ratings. With 3,900 movies and 6,400 movies, there are maximum 23,556,000 rating. According to 1,000,000 ratings, the percentage of rating matrix is 4.25%. It means that 1000 users who saw movies, there are 4.25 users made rating.

The dataset is converted to this experiment, the result consists of 1,000 users and 877 movies. There are 336,736 ratings and each collected user has rated at least 20 movies. The principles of conversation are finding the movies that have the largest rating and the users have many ratings as possible. Rating number is an integer that lies between 1 and 5. Users who rated greater number means they prefer more. This

data set also includes basic demographic information of all users such as age, gender, and occupation.

c. Preprocessing

The structure of dataset is:

The text file "users.dat" follows bellow format

UserID::Gender::Age::Occupation::Zip-code

Voluntarily, all users provided their demographic information and they are checked for validation. All ratings must be rated from users who have their personal information, it is not from anonymous guests.

Gender: M for male, F for female

Age is following bellow table

Classification	**Age**
1	<18
18	18_24
25	25-34
35	35-44
45	45-49
50	50-55
56	>56

Table 5 Age classification in the experiment

Occupation is following bellow table 7

ID	**Occupation**
1	academic/educator

ID	**Occupation**
2	artist

3	clerical/admin	12	programmer
4	college/grad student	13	retired
5	customer service	14	sales/marketing
6	doctor/health care	15	scientist
7	executive/managerial	16	self-employed
8	farmer	17	technician/engineer
9	homemaker	18	tradesman/craftsman
10	K-12 student	19	Unemployed
11	lawyer	20	Writer

Table 6 Occupation classification in the experiment

The text file "movies.dat" follows bellow format

MovieID::Title::Genres

Genres includes follow this list:

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

Genres field are pipe-separated for containing many genres in one filed.

The text file “rating.dat” consists of

UserID::MovieID::Rating::Timestamp

These text files are imported to excel for preprocessing. From excel, dataset is export to Microsoft SQL Server 2008. Figure 11 shows the relationship in the relational database management system.

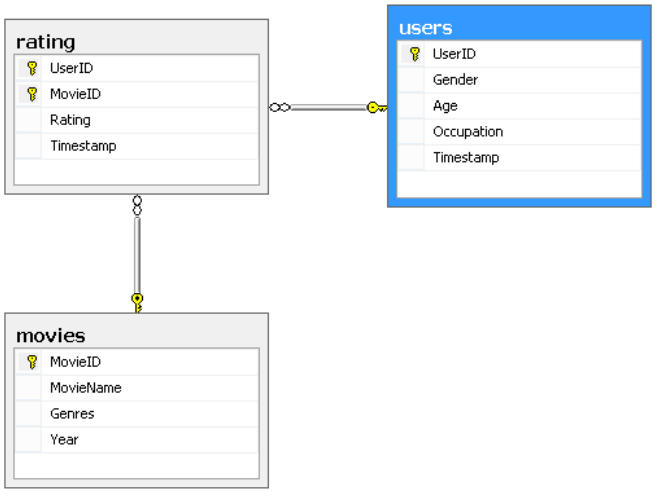

Figure 5 Relational database for the experiment

d. Application interface

This section shows the program interface based on proposed framework. Figure 12 shows the home page of the program without recommendation.

Figure 6 Home page of experiment website

In the same home page, popularity based recommendations are presented in figure 13 “The most popular movies” and “Top highest rating movies”

The most popular movies

MovieID	Movie Name	Genres	RatedUsers	Rating
1196	Star Wars: Episode V - The Empire Strikes Back (1980)	Action|Adventure|Drama|Sci-Fi|War	901	4.36
260	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Fantasy|Sci-Fi	865	4.52
1270	Back to the Future (1985)	Comedy|Sci-Fi	848	4.06
1580	Men in Black (1997)	Action|Adventure|Comedy|Sci-Fi	838	3.68
1210	Star Wars: Episode VI - Return of the Jedi (1983)	Action|Adventure|Romance|Sci-Fi|War	832	4.07
2571	Matrix, The (1999)	Action|Sci-Fi|Thriller	828	4.25
589	Terminator 2: Judgment Day (1991)	Action|Sci-Fi|Thriller	825	4.02
480	Jurassic Park (1993)	Action|Adventure|Sci-Fi	822	3.77
608	Fargo (1996)	Crime|Drama|Thriller	811	4.33
2858	American Beauty (1999)	Comedy|Drama	810	4.37

Top highest rating movies

MovieID	Movie Name	Genres	RatedUsers	Rating
858	Godfather, The (1972)	Action|Crime|Drama	717	4.6
1198	Raiders of the Lost Ark (1981)	Action|Adventure	809	4.55
260	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Fantasy|Sci-Fi	865	4.52
318	Shawshank Redemption, The (1994)	Drama	731	4.49
50	Usual Suspects, The (1995)	Crime|Thriller	680	4.48
527	Schindler's List (1993)	Drama|War	698	4.47
912	Casablanca (1942)	Drama|Romance|War	555	4.44
750	Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1963)	Sci-Fi|War	541	4.42
1193	One Flew Over the Cuckoo's Nest (1975)	Drama	560	4.42
1221	Godfather: Part II, The (1974)	Action|Crime|Drama	603	4.39

Figure 7 The most popular and top highest rating movies recommendations of the experiment website

In figure 14, the system finds “Similar movies” by using correlation of viewed item and other items to create a list of items that have highest similarity. Furthermore, the list of “People also watch” is created by association rules. Figure 14 shows the detailed page of a web session when users click on movie “Movie Name: Godfather, The (1972)”

Movie Name: **Godfather, The (1972)**
Genres: **Action|Crime|Drama**
Average Rating: **4.6**
Number of rated users: **717**

Similar movies...

MovieID	Movie Name	Genres	Similarity (%)	Rating
1221	Godfather: Part II, The (1974)	Action|Crime|Drama	72	4.39
111	Taxi Driver (1976)	Drama|Thriller	33.01	4.14
1032	Alice in Wonderland (1951)	Animation|Children's|Musical	32.34	3.77
1276	Cool Hand Luke (1967)	Comedy|Drama	32.03	4.19
1213	GoodFellas (1990)	Crime|Drama	31.49	4.31
1350	Omen, The (1976)	Horror	30.33	3.54
3468	Hustler, The (1961)	Drama	29.63	4.17
1084	Bonnie and Clyde (1967)	Crime|Drama	29.35	4.09
3098	Natural, The (1984)	Drama	28.66	3.79
913	Maltese Falcon, The (1941)	Film-Noir|Mystery	28.61	4.32

People also watch...

MovieID	Movie Name	Genres	Rating
1196	Star Wars: Episode V - The Empire Strikes Back (1980)	Action|Adventure|Drama|Sci-Fi|War	4.36
260	Star Wars: Episode IV - A New Hope (1977)	Action|Adventure|Fantasy|Sci-Fi	4.52
608	Fargo (1996)	Crime|Drama|Thriller	4.33
1270	Back to the Future (1985)	Comedy|Sci-Fi	4.06
1198	Raiders of the Lost Ark (1981)	Action|Adventure	4.55
593	Silence of the Lambs, The (1991)	Drama|Thriller	4.35
1210	Star Wars: Episode VI - Return of the Jedi (1983)	Action|Adventure|Romance|Sci-Fi|War	4.07
589	Terminator 2: Judgment Day (1991)	Action|Sci-Fi|Thriller	4.02
2571	Matrix, The (1999)	Action|Sci-Fi|Thriller	4.25
1097	E.T. the Extra-Terrestrial (1982)	Children's|Drama|Fantasy|Sci-Fi	3.99

Figure 8 An example of similar movies recommendations