International Journal of u-and e-Service, Science and Technology
Vol.7, No.5 (2014), pp.15-26
http://dx.doi.org/10.14257/ijunnesst.2014.7.5.02

User-Advertisement Simulation: An Approach for Measuring the Accuracy of Collaborative Recommender Systems without Dataset

Niki Shakeri[1], Jinan Fiaidhi[2], Sabah Mohammed[3] and Tia-hoon Kim[4]

[1,2 & 3]*Dept. of Computer Science, Lakehead University, Thunder Bay, ON, Canada*
[4]*Department of Convergence Security, Sungshin W. University, Korea*
nshakeri@lakeheadu.ca, jfiaidhi@lakeheadu.ca, mohammed@lakeheadu.ca and
taihoonn@sungshin.ac.kr

Abstract

Recommender systems are now popular both commercially and in the research community, where many approaches have been suggested for providing recommendations. However, evaluating the affectivity of recommender systems is a challenging problem and most of the approaches used for evaluation are based on using some sort of a dataset. This paper describes a method for measuring the accuracy of a collaborative filtering based recommender systems called "User-Advertisement Simulation" that utilizes a simulation approach that creates artificial users and advertisements of a virtual market, then measures accuracy of the products' ranking based on the user's profile.

Keywords: *Recommender Systems, Collaborative Filtering, Evaluation Techniques*

1. Introduction

Many modern applications are using recommender systems and it makes them able to expose the user to a wide collection of items. Typically these systems provide the user with a list of recommended items they might prefer, or predict how much they might prefer each item. These systems are very helpful in terms of decision making and ease the task of finding preferred items in the collection [5].

For instance, the displayed books in Amazon are provided with the average user ratings and a list of other books that are bought by users who buy a specific book. The online DVD rental provider Netflix shows predicted ratings for all the TV series and movies based on the user's profile. Also Microsoft offers many free downloads for users and once a user download something, the system recommends other software which might suite the user. Although these systems provide a different range of products, all of them are categorized as recommender systems [5].

There has been a huge amount of research in the recommender systems field in the past decade. The main focus of that research has been new algorithms for recommendations. A designer of a new recommender system should go through a large amount of algorithms and find the most appropriate one suitable with the purpose of the system. In general, these type of decisions are based on experiments and comparing the performance of other major recommenders. Researchers, who suggest new recommendation algorithms, will evaluate the performance via available data sets or by applying some evaluation metric that provide ranking of candidate algorithms.

In some applications, the ranking algorithm strongly depends on both user and product's profile. These type of recommenders try to modify those characteristics over time and based on the ratings provided by users. In that context, typically after publishing each rating, the characteristics of the user and advertisement should be modified instantly. One of the very

ISSN: 2005-4246 IJUNNEST

good examples of this type of recommender systems is “Meta Broker”. Meta Broker is a recommender system which provides the users with a ranked list of real estate advertisements, based on their query and using both user-advertisement characteristics. This application has a new approach for modeling the user-advertisement profile and it has no data set to be evaluated with [6].

In this paper we propose a new approach for evaluating recommendation system, called “User-Advertisement Simulation”. Then Meta Broker will be evaluated by this method and results will be discussed.

2. Overview of the User-Advertisement Simulation

Having the appropriate results is the first step to be done for all the recommender systems. However, it does not indicate if the ranking of the products is the best possible fit for the user. For this purpose a Simulator has been developed to measure the accuracy of the rankings per user.

The User-Advertisement Simulator is going to create an environment involving users and advertisements to measure the accuracy of the program. For preparing the simulated environment, artificial users and advertisements will be added. The users and advertisements profile are generated and will be kept as a measurement variable but in the meantime, they are inserted to the simulation as unknown characters. Afterward the simulations picks two candidates from users and advertisements and the rating will be published automatically. The original recommender system is still responsible for updating the affected characters after each feedback from the users. A measurement parameter will be defined and the simulation gets sample and calculates the current parameter. The mentioned parameter represents how close the recommender system is to the ideal form and by using it, User-Advertisement Simulation publishes some charts to illustrate the progress of the original program over time and based on ratings.

The following secession explains the structure of the simulation in details and how it will be applied to Meta Broker as a collaborative recommender system.

3. Architecture of The User-Advertisement simulation

As it is shown in Figure 1, the simulation starts with generating Golden Users and Advertisements.

The Golden Users and the Golden Advertisements are artificial users and advertisements in which each field of their respective characteristics are generated randomly. So, this ensures that all of values for the characteristics of the Golden Users and the Golden Advertisements are known. Then, the Golden Users/Advertisements are inserted into the Simulator as new items, therefore their known characteristics are altered so that they have values of unknown. The unknown initial values for building profiles up, might be different among recommender systems. Meta Broker as our sample uses 50/100 for the initial value of all the characteristics.

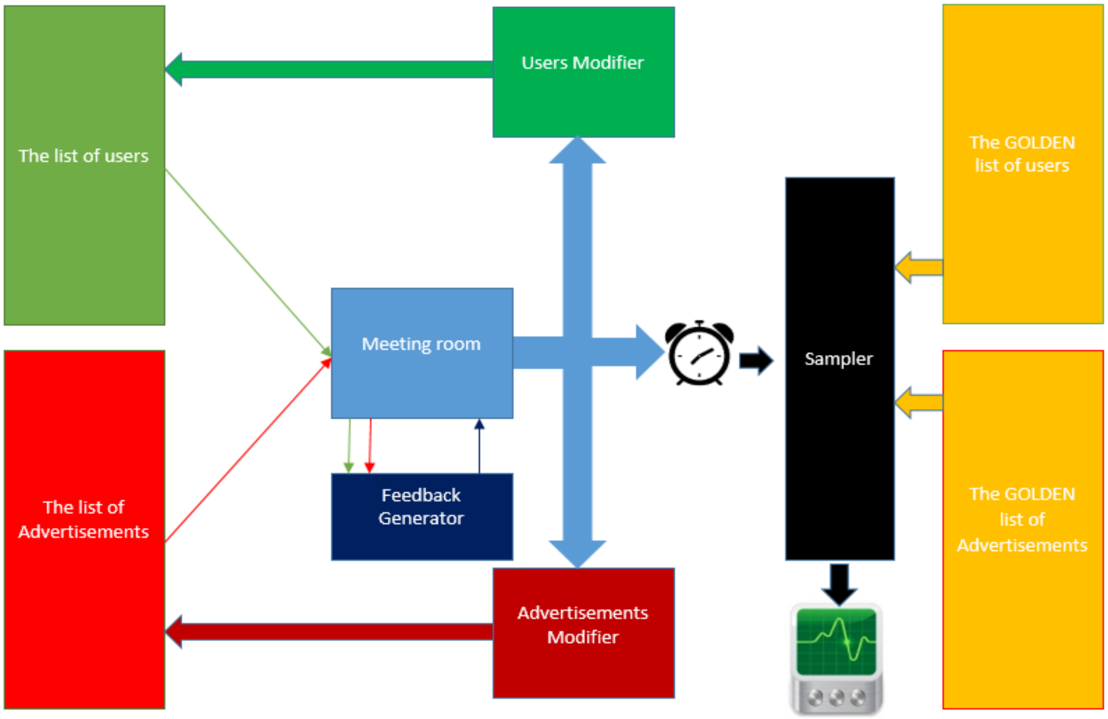


Figure 1. The Structure of the User-Advertisement Simulator

Next, some of the inserted items (no longer Golden because their characteristics were reset to 50) are selected by the Meeting Room. The Meeting Room connects a user to an advertisement so that the user can publish feedback on the selected advertisement in the Feedback Generator. The Feedback Generator creates feedback with the characteristics provided by the Golden Users and the Golden Advertisements. This is like a real human, with a known consumer behaviour, giving feedback on an advertisement in which the advantages and disadvantages of the advertisement are known. Then, the user and the application modifiers will update their candidates' information. These two components, the User Modifier and the Advertisement Modifier, are completely identical to what the original recommender system uses. This procedure is repeated for a while and after a certain time The Sampler gets a snap shot from the progress of the system which allows it to compare the list of users and advertisements characteristics with the original Golden Lists.

There is a parameter which will be measured to see how close the rankings are to the best fit. Figure 2 is an illustration of the pseudo code of the Simulator's program.

```
Generating the Golden list of users
Generating the Golden list of advertisements
Inserting all the new users and advertisements into the Simulator
While (Sample counter < Limit) {
        A pair of user-advertisement will be picked by the Meeting Room
        A user publishes feedback on an advertisement through the Feedback Generator
        User & Advertisement Modifier modify their item
        Sample counter ++
        If (sample counter % sample rate = 0) {
                Compare the rankings with the Golden ranking
                Publish the comparison
        }
}
```

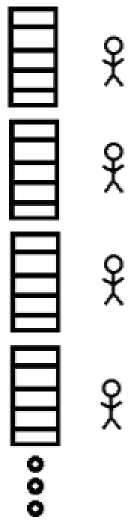
Figure 2. Pseudo Code of The Simulator's Functionality

The Sampler is responsible for comparing the current results with the best fit. Also, for measuring the accuracy of the ranking, an Optimization Parameter (OP) that represents the performance which the recommender system has been developed. The following sections elaborate for a better understanding of the functionality of the Simulator, the mentioned parameter, Meeting's Room, and the Feedback Generator.

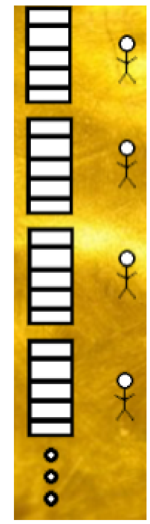
3.1. Optimization Parameter

If we suppose that the number of users is U and the number of advertisements is A, each ranking for any user contains A items. Then all the rankings are merged into a list with I=A x U items, which is called the Final Ranking (FR) list, and where I is the number of items in the FR list. With the usage of both Golden lists, the Golden Final Ranking (GFR), with I items can be created. Also, at any time it is possible to generate the Final Ranking list.

A snap shot is when this list is generated with the current characteristics from both the users and the advertisements. Thus, whenever the sampler takes a snap shot, the Final Ranking will be generated and then it will be compared with the Golden Final Ranking. Both lists have I items and each item has an index. Equation 1 defines the Optimization Parameter, which indicates the accuracy of the sample. Firstly, the differences between the two lists, the FR list and the GFR list, for each user need to be added. Then it is divided by A squared because the maximum of each difference is A and there are A items for each individual. So, all the users will come up with a number and the next step would be getting the average among all the users.



$$OP = \frac{\sum_{1}^{U} \frac{\sum_{1}^{A}(|Sj - Gj|)}{A^2}}{U}$$



Eq. 1: Optimization Parameter Formula

The maximum value of the Optimization Parameter, is 1 and the minimum value is 0. The lower values show that the ranking is getting closer to the ideal or the GFR. Figure 3 shows the pseudo code for the sampler component.

```
Get the Golden ranking GFR, G
Total = 0
For each user as i {
        Get the ranking FR, S
        U[i]=0
        For each advertisement as j {
                U[i] + =|Gj-Sj|
        }
        U[i]/=A x A
        Total+=U[i]
}
Total/=U
Publish Total as the current state
```

Figure 1. Pseudo Code of the Sampler Component

3.2. Meeting's Room

A component responsible for connecting users with advertisements is called "Meeting's Room". User-Advertisement Simulation has included three structures for the Meeting's Room. Each one of them is an approach managing how users should publish ratings over advertisements. These three methods are called Random Connections, User's based Connections, and Advertisement's based and will be discussed as follow.

3.2.1. Random Connections

This method is very similar to what happens in reality, because only some of the users publish feedback on some of the advertisements. The Meeting Room component pairs a user and an advertisement, then the user has to publish feedback on that advertisement. The total number of all the possible connections is A x D and the Simulator targets a part of it using the rate parameter. Figure 4 is the illustration of the structure of Random Connections and Figure 5 shows the pseudo code of this method.

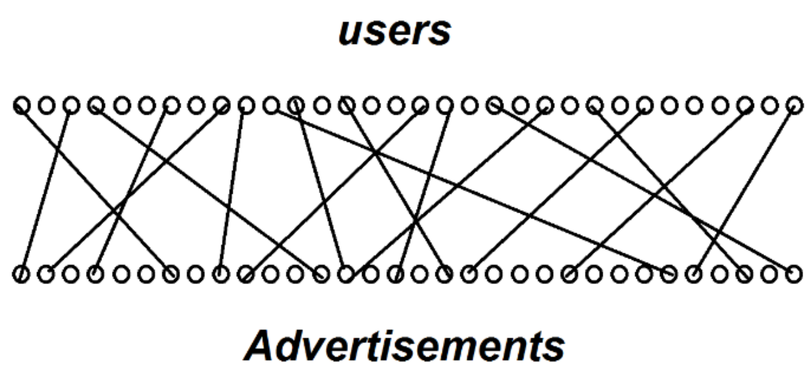


Figure 5. Pseudo Code for Random Connections

```
Total = A x D
Rate = 10%
Limit = Total x Rate
Count=0
While (count < limit) {
        Pick a random non redundant pair of (U, A)
        Get the feedback from the user on the advertisement
        Count++
        If (count%100 = 0) {
                Regenerate the OP
                Publish the OP
        }
}
```

Figure 4. Structure of Random Connections

3.2.2. User's based Connections

The Users' based method, as the name suggests, focusses on the users. Each user has to publish feedback on all of the advertisements in turn. Figure 6 shows the structure of the Users' based method and Figure 7 explains the pseudo code of it.

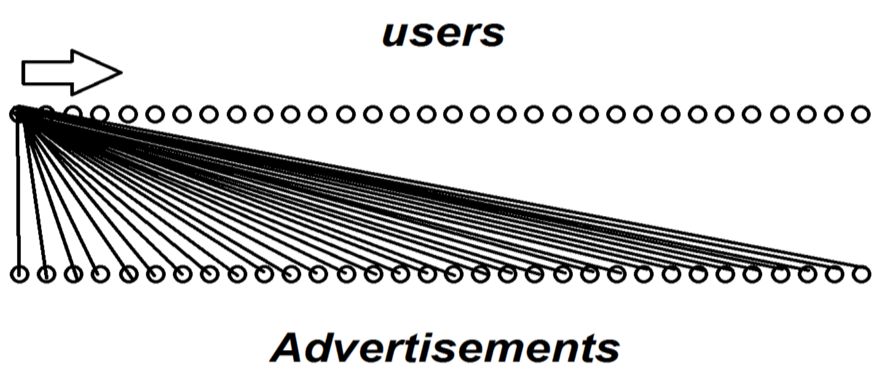


Figure 6. Users' Based Structure

```
For each users {
        For each Advertisements {
                Get the feedback from the user on the advertisement
        }
        Regenerate the OP
        Publish the OP
}
```

Figure 7. Pseudo Code of the Users' Based Connections

3.2.3. Advertisement's based Connections

This method is exactly the reverse form of the users' based. In this method advertisements are fixed and users are variables. Figure 8 shows the structure and Figure 9 describes the pseudo code for it.

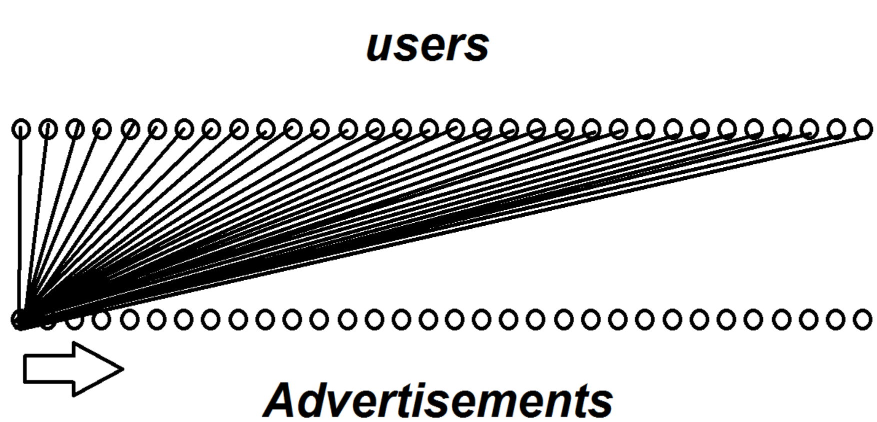


Figure 8. Advertisement's Based Connections Structure


```
For each Advertisements {
    For each users {
        Get the feedback from the user on the advertisement
    }
    Regenerate the OP
    Publish the OP
}
```

Figure 9. Pseudo Code of the Advertisement's Based Connections

3.3. Feedback Generator

The Feedback Generator is another influential component. We believe any given feedback is the combination of the user and advertisement. This means there is not any other variable involved in the process of publishing feedback. In reality a human should give feedback on a real advertisement but we try to model that situation with using Golden Vectors of both users and advertisements. As it is shown in Equation 2, the feedback generator uses a linear formula which considers noise. In this formula UC represents the user's characteristics for a specific field and similarly, AC represents the advertisement's characteristics.

Also α represents a possible noise which is a randomly generated number from -5 to 5.

$$F = AC - UC + 50 + \alpha$$

Eq. 2: The Feedback Generator Formula

If we suppose that UC = 50, the entry value of users' characteristics, we can conclude that F = AC + α, which means an absolutely normal person will see the reality of the advertisement with a little bit of noise. If we suppose all the variables are fixed and UC>50 then we are dealing with a more sensitive user, and F will be decreased. The reason is those types of users can hardly be satisfied so their feedback is less than the normal user's. On the other hand, if UC<50 we see that F will be increased because these types of users are more easy going and they will publish better feedback on the same advertisement. Table 4.1 represents several types of generated feedback using Equation 2, from various types of users on several types of advertisements.

Table 1. The Feedback Generator's Results

UC	AC	α	F
20	10	2	42
20	50	-3	77
20	90	-4	100
50	10	3	13
50	50	0	50
50	90	2	92
80	10	5	0
80	50	0	20
80	90	-1	59

4. Results

By modeling the Meta Broker and running the simulation with three different structures of Meeting's Room, different results have been achieved. The results are the outcome of building the simulation with 1000 users, 1000 advertisements, and different Meeting's Room method.

Figure 10 shows the result of applying 10% of connections on the Simulator when the environment has 1000 users, 1000 advertisements, and 1000 snap shots. At first, all the users and advertisements contain unknown value for all their fields and at that time, almost every advertisement is 45% away from the correct ranking position.

Then, when the number of connections reaches 1%, Meta Broker's OP gets close to around 3% and stays there forever. This shows that Meta Broker is a fast learner because it was able to get a very accurate OP with only 1% of connections. This means that if there are 100 possible advertisements, and based on a user query an advertisement should be ranked in the 35th position, one can expect to see that advertisement ranked between the 32nd to 38th positions.

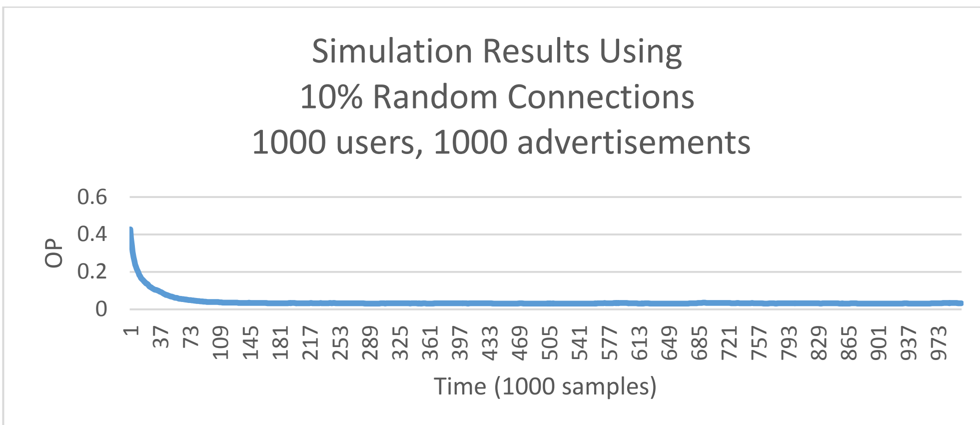


Figure 10. Simulation Results for 10% Random Connections (U=1000, A=1000)

By changing the meeting's room method from Random Connections to User's Based Connections, the result is very interesting (see Figure 11). All of the learning happens when the second user has finished publishing feedback. The reason for this phenomenon is that after the first user publishes feedback for each advertisement, that user becomes mature, and his/her feedback is trustworthy. Therefore, when the second user publishes his/her feedback the Simulator has already acquired trustworthy feedback from the first user and thus the Simulator is better able to understand the second user's feedback and in turn rank the advertisements.

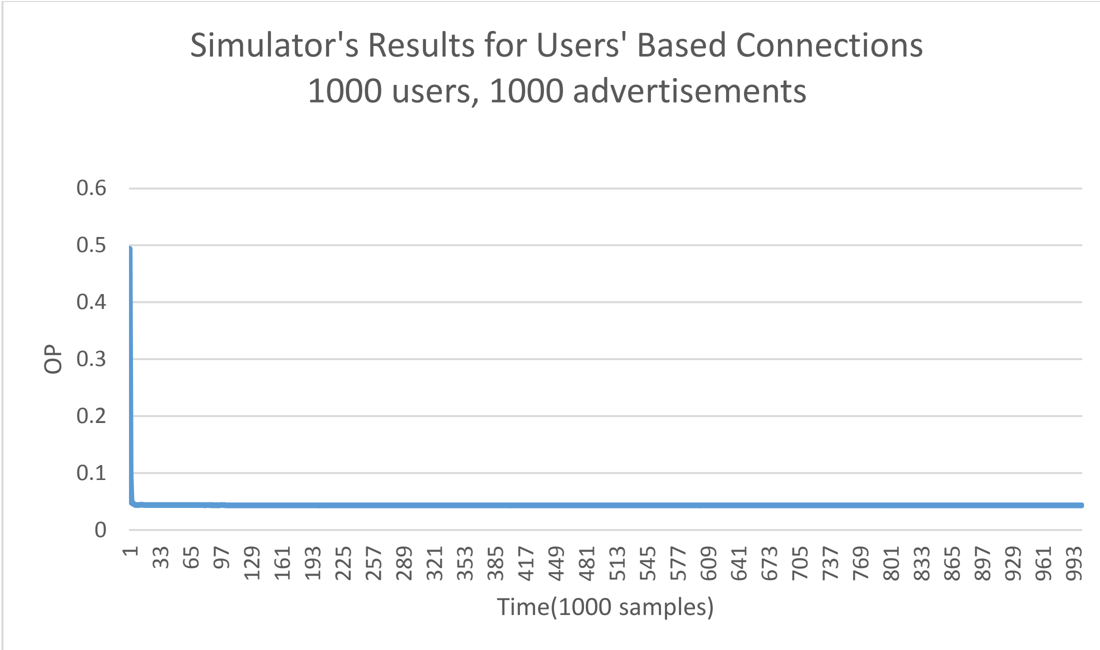


Figure 11. Simulation Results for Users' Based Connections (U=1000, A=100, S=1000)

Once we change the Meeting's Room method to Advertisement's based and get the result, analysis show the chart starts with an Optimization Parameter of 50% and reaches 3% OP at the end, see Figure 12. There is not any sign of saturation, which shows that Meta Broker is constantly learning. The reason for the difference between users' based and advertisements' method is that in users' based, advertisements are given feedback immediately while the majority of users are immature for a while. This combination is able to get a ranking close to the golden one.

However in the Advertisements' based method the majority of users become mature and advertisements will be kept immature. So, when most of the advertisements have not received critique, Meta Broker is not able to learn about them quickly.

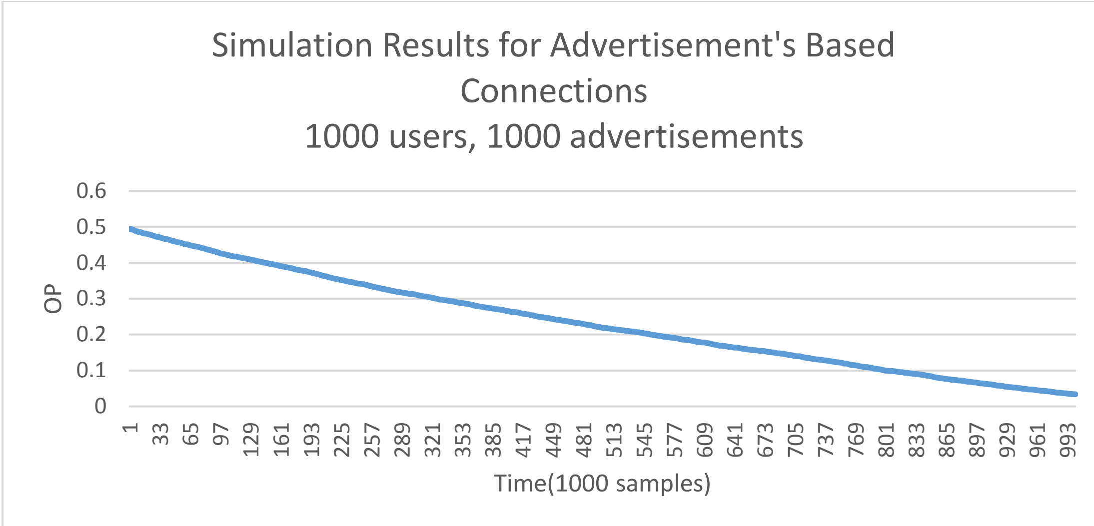


Figure 12. Simulation Results for Advertisements' Based Connections (U = A = S = 1000)

Table 2 shows the comparison of all the mentioned methods with 1000 samples. This test shows an environment close to reality, Meta Broker is able to reach the 3% OP with just 1% of the connections. This result proves how accurate and quickly Meta Broker can learn.

Table 2. Random Connections/Users' based/Advertisements' Based Comparison

Method	Starting Op	Saturation OP	Saturation Sample	Drop OP	Drop Sample
Random Connections	45	3	1	10	0.05
User's Based	48	3	0.02	3	0.02
Advertisement's based	47	3	10	N/A	N/A

5. Discussion and Conclusion

In this paper, we presented a prototype for a simulator to evaluate a collaborative recommender system. This simulator is able to build a virtual market and try to match user's requirements with the available advertisements according to the user automatic feedback. The simulator prototype provided an effective way to measure the effectivity of the recommender system. The simulator was able to measure the accuracy in various levels and adjust the best possible way to hit the highest accuracy possible. The simulator highest accuracy reached was of 97%. We are still in the process of developing a more advanced version which applies more intelligent algorithms for building initial artificial users/advertisements and the generating feedbacks from users, to be more compatible with the natural consumer behavior.

References

[1] M. Klusch and K. Sycara, "Brokering and matchmaking for coordination of agent societies: a survey", In Coordination of Internet agents, Springer-Verlag, **(2001)**.
[2] J. Fiaidhi, S. Mohammed and M. Hahn, "Developing a Brokering Architecture for Multimedia Learning Objects on the Semantic Web", IJCSNS Int. Journal of Computer Science and Network Security, vol.7, no.1, **(2007)**.
[3] P. Karthikeyan and E. Sathiyamoorthy, "A Survey on Applications of Mobile Agents in E-Business ", International Journal of Scientific & Engineering Research, vol. 3, no. 3, **(2012)**.
[4] L. L. Geng, "Study of e-commerce system based on mobile agent", IEEE International Conference on Computer Science & Information Technology (ICCSIT), **(2010)**; Chengdu, China.
[5] G. Shani and A. Gunawardana, "Evaluating recommendation systems", **(2011)**.
[6] J. Fiaidhi, N. Shakeri, S. Mohammed and T. H. Kim, " Collaborative Filtering Methods for Identifying Relevant Adverts to a Real Estate Mobile Agents", International Journal of u- and e- Service, Science and Technology, IJUNESST, vol. 7, no. 4, **(2014)**.