COLLABORATIVE FILTERING RECOMMENDATION SYSTEM: A FRAMEWORK IN MASSIVE OPEN ONLINE COURSES

D.F.O.Onah[1*], J.E. Sinclair[1]

[1]The University of Warwick (UNITED KINGDOM)

ABSTRACT

Massive open online courses (*MOOCs*) are growing rapidly in the educational technology environment. There is a need for *MOOCs* to move away from its one-size-fit-all mode. This framework will introduce an algorithm based recommendation system, which will use a collaborative filtering method (*CFM*). Collaborative filtering method (*CFM*) is the process of evaluating several items through the rating choices of the participants. Recommendation system is widely becoming popular in online study activities; we want to investigate its support to learning and the encouragement to more effective participation. This research will be reviewing existing literature on recommender systems for online learning and its support to learners' experiences. Our proposed recommendation system will be based on course components rating. The idea was for learners to rate the course and components they have studied in the platform between the scales of 1 – 5. After the rating, we then extract the values into a comma separated values (*CSV*) file then implement recommendation using Python programming based on learners with similar rating patterns. The aim was to recommend courses to different users in a text editor mode using Python programming. Collaborative filtering will act upon similar rating patterns to recommend courses to different learners, so as to enhance their learning experience and enthusiasm.

Keywords: recommendation, collaborative filtering, MOOC, Python, learners, massive open online courses

1 INTRODUCTION

Collaborative learning has been recognised worldwide as an effective way of learning. Collaborative technologies and learning pattern have grown so popular as an advance process of research and learning. Introducing collaborative learning methodology helps to elevate MOOCs from its one-size-fit-all concepts of learning. One of the popular technologies applied in the past and in recent studies is the recommender system. Recommender systems have proven to be a diligent application concept in online learning for some time. These learning patterns facilitate the sharing of knowledge or collaboration between similar learning styles amongst the participants. These systems provide users with proactive and personalised learning patterns [1]. Collaborative filtering (*CF*) techniques have been proven to be one of the most useful facilitator of the recommender systems by providing an opportunity for users to rate the course concepts on how suitable and important they are to their learning preferences. We consider using this *CF* technique in our current course development, to investigate how efficient it will be in supporting learning, and reducing the high dropout rate in online learning systems especially Massive Open Online Courses (MOOCs).

According to some authors' reviews, recommender systems have been applied in several virtual domain intelligent systems [1, 2, 3,4]. User profile in combination with the participants information filtering system have been recognised as one of the most important components within recommender systems to enable effective delivery of appropriate learning contents suitable to learner's preferences and desires. There are two commonly used recommender strategies:

Content-based recommender strategy: This comprises the description of the course contents or items to be recommended, which are then compared to the learner's preferences [5].

Collaborating filtering (CF) strategy: This strategy of *CF* recommendation provides solution by relying on the users profiles, past learning or rating histories of the previous and current learners [3, 6]. The *CF* recommendation is done based on similar rating history and patterns of learning behaviours exhibited by the users.

In this research, we are sorely interested in collaborative filtering strategy in our framework, and how it can be applied by using similar rating patterns to recommend course contents to learners, which are suitable to their learning preferences. Though, on the other hand, we are not ignorant of the fact that trustworthiness within similar ratings, sometimes according to some authors will not be accurate to conclude and judge the prediction of the best course concepts to be recommended to specific user profile [1, 7, 8]. However, according to *O'Donovan* "trustworthiness of a partner should be considered". The issue with measuring trustworthiness within *CF* based on similar rating history have been argued by [9,10], taking into consideration the social and technological implications on the participants. In a similar experimental report by *Golbeck et al.,* their evidence reported that bad nodes within the trustworthiness recorded in the learners network causes their " rating accuracy to drop drastically"[11].

In another note, collaborative system offer personalised recommendation, which provides contents to the users based on their potential interest and similarities with the rating history of different learners [12,13,14]. As highlighted by *Lin et al.,* [15], despite the awareness of recommendation technology system, most techniques known are readily dependable on simple methods to measure and represent learning similarities within the participants, particularly in "the linear correlation coefficient for a given pair of users".

In the system implementation, we will be extending on existing prediction functions as part of the concepts to implement in our proposed framework using Python programming algorithm for the *CF* recommender system. The functions are briefly discussed in this paper. However, we are still in the working process to fully understand the accuracy of the algorithm. We intend to make the full algorithm and concepts published after the successful implementation of the system in our further research.

This research is structured as follows: firstly, a review of literature on recommendation system and techniques. Secondly, discussion of our proposed collaborative filtering approach using a text editor in Python programming and some existing functions. We conclude with discussion of the proposed system architecture framework, and suggestion of further research progress and directions.

2 LITERATURE REVIEW

Several recommender systems are in existence nowadays and have been applied in learning and e-commerce in the past. Collaborative filtering (*CF*) techniques have been seen as one of the most widely applied methods to offer recommendations to users by matching the participants taste and interest to like-minded learners and recommend contents accessed, and read by the others as recommendations [16, 17. 18]. Recommender system according to *Ujjin et al.,* [19] is one way or "circumventing" the problems of choices amongst users. The task as highlighted by *Ujjin et al.,* is to recommend or suggest contents based on the participants preferences. Several authors pointed out that recommender systems (*RSs*) are software application tools and techniques applied into making suggestions for certain item or contents to be delivered to particular users based on their decisions [20,21,22,23].

2.1 Recommendation Techniques

There are several recommendation techniques applied in learning system. This session will be discussing some of these techniques and their effectiveness to enhancing learners' experiences.

2.1.1 Collaborative filtering

Collaborative filtering unlike information filtering (*IF*) which deals with the analysis of content and the individual users profile interest. Collaborative filtering focuses on the identification of learners with similar learning patterns and uses their learning methods to recommend contents to others [24]. *Mwlville et al.,* mentioned that most recommendation systems applied collaborative filtering methods or content-based techniques to recommend items of interest to the users, but they have a flaw of failing to suggest better recommendations in most cases [5]. They argued that they combine both content-based and collaborative filtering methods to enhance existing user data and then proffer a personalized suggestion accordingly using their proposed approach of "*Content-Boosted Collaborative Filtering*".

2.1.2 Rated Recommendation System

Rated system is a form of collaborative filtering mechanism where learners rate course contents. After the rating is completed, then a recommendation is done to learners with similar rating patterns. Assuming a new learner begins the course, the system will recommend the general features, as they rate and progresses in the course, then the rating system functionality will be in effect. In contrast to the effectiveness of the rating system, *O'Toole* [25], in a discussion paper about peer assessment, claims that "whereas in the *cMOOC* participants are primarily interested in building the collective capabilities of the whole network, and hence are more likely to use feedback and ratings systems honestly, in *xMOOCs* participants are aiming to get a good personal grade".

2.1.3 Association Rule Mining

According to *Lin et al.,* who explained how they investigated association rule mining as an underlying technology for collaborative recommender system [15]. *Fu et al.,* argued that they have developed a system that recommends web pages using *'Apriori algorithm'* as a domain recommender system to mine association rules applied to users' navigation histories [26]. *Sarwar et al.,,* however, described using a traditional rule mining approach in his work by pre-selecting reduced number of collaborative learners who are closely related to the target learner for content recommendation [27]. However, *Lin et al.,* argued that most association rules were not altogether suitable for some of the intended domain, the reason is that they did not provide mechanism to support the minimum confidence and the desired range involved in the available rules [15]. One of the criticism about the association rule mining according to *Lin et al.,* was that it leads to either too many or fewer rules, which takes lots of computation period or poor performance classification. However, having mentioned that, *Webb* [28] and *Agarwal et al.,* [29], argued they have addressed some of these criticisms by using judicious search techniques.

2.1.4 Knowledge-Based Recommendation System

This system recommends contents based on specific domain knowledge on the usefulness of the contents to the learners needs and preferences [19]. According to *Bridge et al.,* [30], and *Ricci et al.,* [31], they argued that knowledge-base systems are *case-based systems* which uses a similarity function to access the needs of the learners and provide recommendations. Another example of knowledge-based system is a constraint-based system, which collect learners' requirement and re-adjust the preferences for consistency, and also automatically suggest recommendation if none is offered originally. While case-based recommenders proffer recommendation based on similarity matrices, constraints-based recommenders exploit predefined knowledge bases with explicit rules to suggest contents to learners suitable to their learning needs [19]. However, *Ujjin et al., [19],* argued that knowledge-based systems tend to be more efficient as compare to the others at the early stages or phases of their application, but there exists some criticisms, which they argued that if knowledge-based systems are not properly "equipped with learning components", they certainly can be overshadowed by other methods such as collaborative filtering which can exploit both human and computer interactions.

3 PROPOSED CF FRAMEWORK ARCHITECTURE

Our proposed collaborative framework is to be applied on MOOCs program. We are developing the framework which will apply some of the concepts explained in the literature review. Learners will be able to rate any course they study from scales of 1-5. This rating will be stored in a CSV file and later exported to a directory created using terminal. This file is then accessible from Python environment editor, and then we perform the recommendations using a collaborative filtering algorithm developed in Python language and applying existing prediction function methodology (as seen in figure 1).

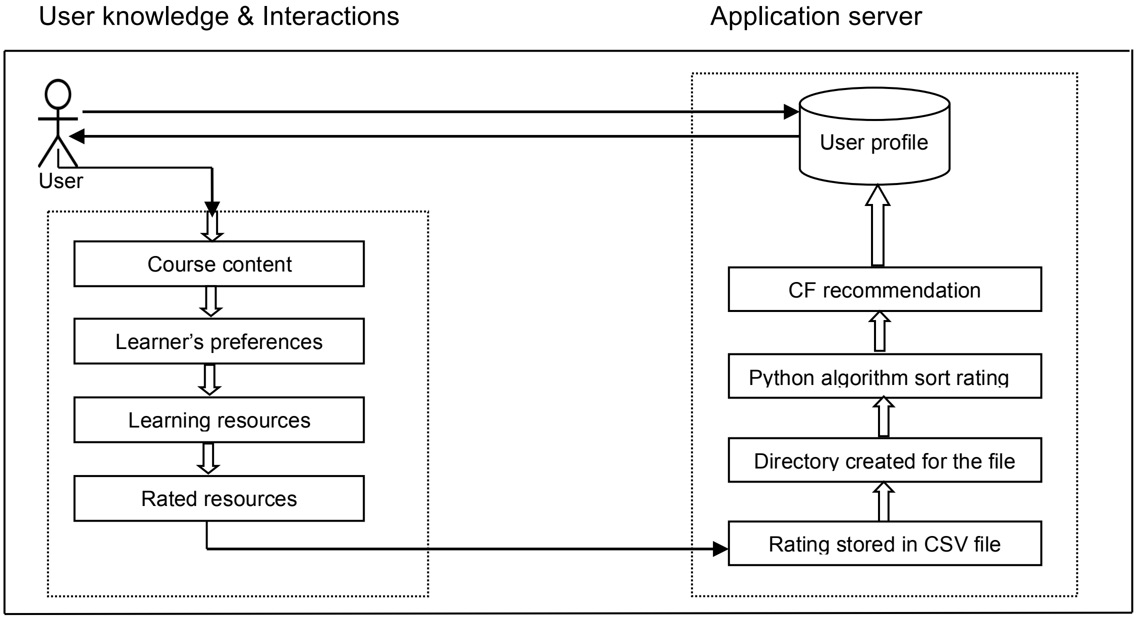


Figure 1. A proposed CF Architecture

3.1 Features of CF Proposed System

The process involved in this architecture is to be able to recommend personalized preference course contents to learners using rating system functionalities. The contents rated are stored in a *CSV* file format, which later stored in the format suitable for Python environment to extract using the *CF* algorithm already developed. A directory file will be created using terminal environment as in the *CSV* file format. The course contents recommendations are done based on the unique users stored profile in the database (as shown in figure 1). Our proposed system will expand on the prediction function to recommending contents to new target user as addressed in [32].

3.1.1 Prediction Function

The target user requesting for recommendation with a set of collaborative users. Using similarity matrix to find a content recommendation for the target user from the collaborative users. The identification of all the collaborative users with similar rating patterns is significant. The prediction function *K*, calculates the prediction of user *t* for content *j* . *p* is the set number of collaborative users, *sim(t,i)* is the collaborative user set for similarity with the target user, then multiply by the rating difference of any of the contents or sub-contents *j* within the whole rating process of the collaborative user. Then sum it up and multiply by the modulation factor *q* and added with the average rating n_t of our target user which will then produce the prediction rating for our target user recommendation [32].

$$K_{t,j} = n_t + q * \left(\sum_{i=1}^{p} sim(t,i) * (n_{i,j} - n_i) \right)$$

The prediction function *K* process select the target user *t* and select all the set of similar collaborative users and offer similar contents of interest to the learners which is seen in the far right equations after the equality sign, that is; n_t plus *q** and the brackets. According to *Basilico et al.,* collaborative filtering and content-based recommendations are two common paradigms in existence for ‘context recommendation systems and learner preference prediction’ [33]. According to *Wang et al.,* [34], they argued that individual predictors rate items in a different manner. Some individuals rates contents based on preferences to how interesting the contents are to them. So with this practice, some contents can be rated higher than the others, which did not reflect and predict the true values of the contents been rated. They were able to resolve these anomalies and drawbacks of these rating behaviours by normalizing the contents matrix before any prediction is done.

3.1.2 Proposed Collaborative-Filtering Algorithm: In Python

This algorithm is written in Python programming language using the concept of gradient decent methods. The flowchart revealing the process of course content rating and recommending contents based on similarity rating. In the case of a new member, general course content is recommended because they are new and therefore no content has been rated for the features to take effect (as seen in figure 2). The first part of the algorithm (seen in figure 3), reads the *CSV* file and relate the user to the contents and further investigate similarity between the other users that participated in the rating as explained in the prediction function earlier.

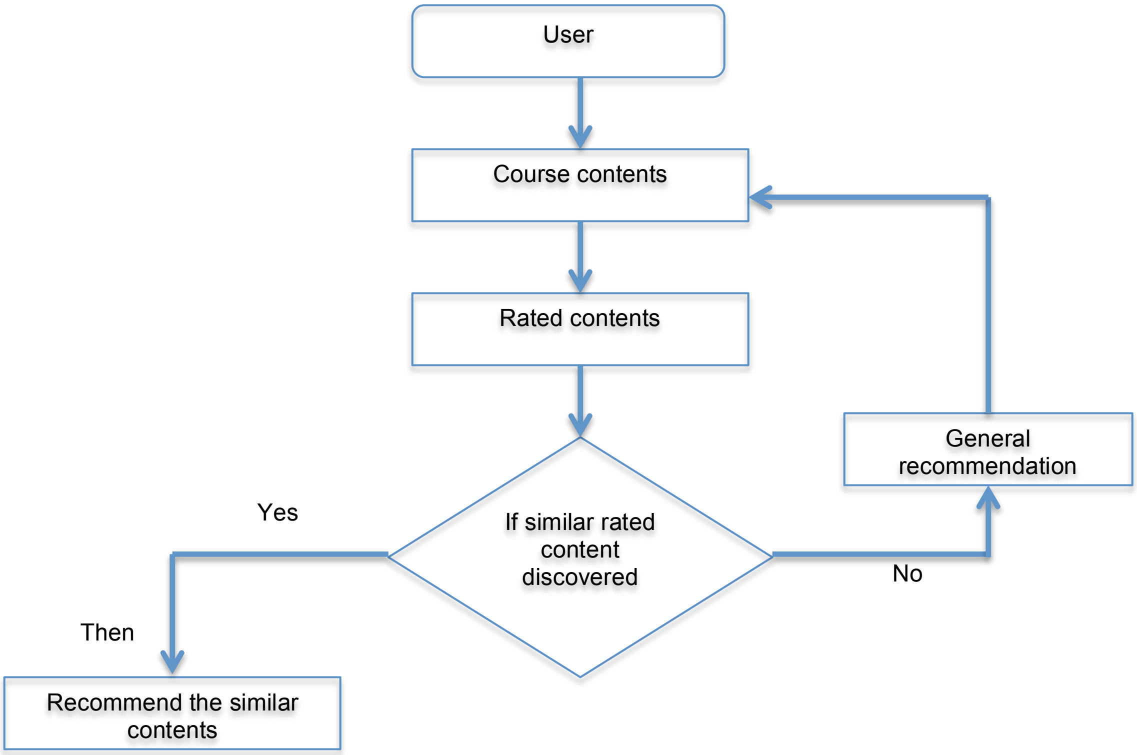
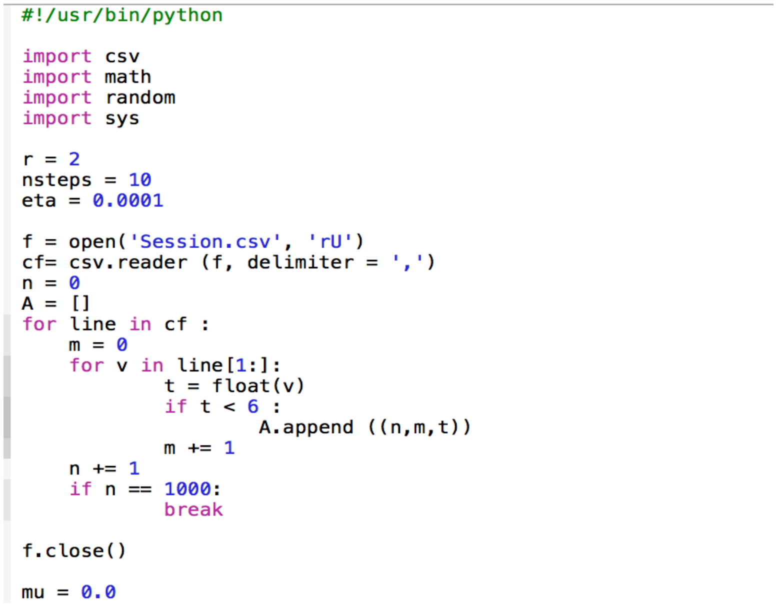

User
Course contents
Rated contents
If similar rated content discovered
Yes
No
General recommendation
Then
Recommend the similar contents

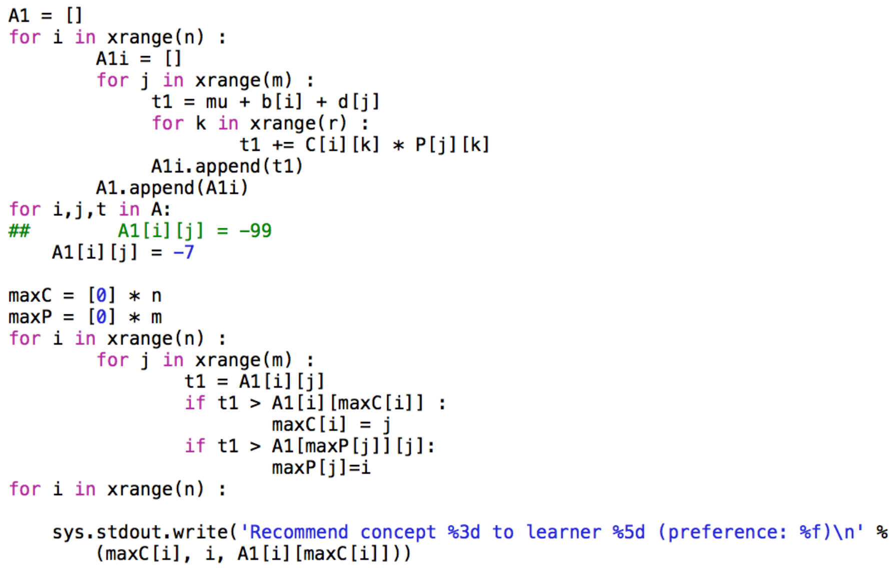
Figure 2. A flowchart of the learner's rating concepts and contents recommendation



```
#!/usr/bin/python

import csv
import math
import random
import sys

r = 2
nsteps = 10
eta = 0.0001

f = open('Session.csv', 'rU')
cf= csv.reader (f, delimiter = ',')
n = 0
A = []
for line in cf :
    m = 0
    for v in line[1:]:
            t = float(v)
            if t < 6 :
                    A.append ((n,m,t))
            m += 1
    n += 1
    if n == 1000:
            break

f.close()

mu = 0.0
```

Figure 3. A screen shot of our working Python algorithm reading the CSV file

The second parts try to calculate the range of similarity within the rating collaborations as seen in a selected code below.



```
A1 = []
for i in xrange(n) :
        A1i = []
        for j in xrange(m) :
             t1 = mu + b[i] + d[j]
             for k in xrange(r) :
                     t1 += C[i][k] * P[j][k]
             A1i.append(t1)
        A1.append(A1i)
for i,j,t in A:
##        A1[i][j] = -99
    A1[i][j] = -7

maxC = [0] * n
maxP = [0] * m
for i in xrange(n) :
        for j in xrange(m) :
                t1 = A1[i][j]
                if t1 > A1[i][maxC[i]] :
                        maxC[i] = j
                if t1 > A1[maxP[j]][j]:
                        maxP[j]=i
for i in xrange(n) :

    sys.stdout.write('Recommend concept %3d to learner %5d (preference: %f)\n' %
        (maxC[i], i, A1[i][maxC[i]]))
```

Figure 4. A screenshot of our proposed algorithm in progress showing similarity recommendation

At the final stage (in figure 4), the system recommends the contents suitable to the learners' preferences. These codes are just a fraction of the algorithm, there are much to these as shown in this paper. Because we are still in the process of developing the algorithm, only the sections relevant to this framework is been shown.

4 CONCLUSION

Collaborative filtering has been referred to as the similar learning pattern: this is considered as the similar rating behavior of learners, which are used for predicting recommendations based on learning similarity amongst participants [12]. Our proposed recommendation framework is not a total guarantee that the learners will follow the structured and guided learning methods provided. In line with this view, one wonders if learners decide to follow their own choices to learning, how effective will these learning patterns be to the experience and enhancement of the educational desire of the participants. The paper review some state-of-the-art in recommendation systems applied in the past. Our proposed system looks into the important concepts to apply in improving the learners' experiences in a more advanced collaborative way. With this in mind, we proposed to begin with *CF* using the knowledge domain of the users and tailor our system to their preferences using similar rating and learning techniques. As highlighted by *Rocha,* recommendation systems pro-actively suggest relevant and related contents to the learners, which they may have been unaware of [35]. Recommender systems contributed in overcoming scattered information overload by proffering a personalized suggestions to each individual learners based on their existing history of similarities or dissimilarities [5].
As mentioned in this paper, there are two prevalent approaches to developing recommender systems, which are content-based, and collaborative filtering. While the formal (content-based) approach presents recommendations by way of representation comparison of contents contained in an item or course to the representation of the contents that are of interest to the learners. On the other hand, the later (collaborative filtering) focuses on the collection of learners' feedback in

the form of content ratings within a given course domain, and exploit the similarities and dissimilarities amongst different learners' profiles so as to determine how to recommend appropriate contents to suit their preferences [5]. However, *CF* have some key advantages over content-based recommender system as addressed in [36], that's the reason why we decided to apply this approach in collaboration with some adaptive personalized concepts in our proposed framework architecture.

In our further research work, we will be addressing adaptive learning system inclusive of collaborative filtering system to enhance the content recommendations to learners based on their learning preferences. The rating recommendation system algorithm in Python is an extension of the gradient descent concepts and prediction functions. These concepts will be applied into the course contents for an adaptive personalized system development in MOOC, so as to capture the collaborative filtering features in the final system implementation.

5 ACKNOLEDGMENT

The first author wishes to acknowledge Mr. Adakole. S. Onah's financial support in his research, and family members and friends for their moral support.

6 REFERENCE

[1] O'Donovan, J., & Smyth, B. (2005). Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces* (pp. 167-174). ACM.

[2] O'Sullivan, D., Wilson, D., & Smyth, B. (2002). Improving case-based recommendation. In *Advances in Case-Based Reasoning* (pp. 278-291). Springer Berlin Heidelberg.

[3] Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence* (pp. 43-52). Morgan Kaufmann Publishers Inc.

[4] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175-186). ACM.

[5] Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI* (pp. 187-192).

[6] O'Donovan, J., & Dunnion, J. (2004). A framework for evaluation of information filtering techniques in an adaptive recommender system. In *Computational Linguistics and Intelligent Text Processing* (pp. 502-506). Springer Berlin Heidelberg.

[7] Massa, P., & Bhattacharjee, B. (2004). Using trust in recommender systems: an experimental analysis. In *Trust Management* (pp. 221-235). Springer Berlin Heidelberg.

[8] Massa, P., & Avesani, P. (2004). Trust-aware collaborative filtering for recommender systems. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE* (pp. 492-508). Springer Berlin Heidelberg.

[9] Abdul-Rahman, A., & Hailes, S. (1998). A distributed trust model. In *Proceedings of the 1997 workshop on New security paradigms* (pp. 48-60). ACM.

[10] Marsh, S. P. (1994). Formalising trust as a computational concept. Available at: https://dspace.stir.ac.uk/handle/1893/2010 - .VLV-AFrfjdk [Accessed: 13 January 2015].

[11] Golbeck, J., & Hendler, J. (2004). Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In *Engineering knowledge in the age of the semantic web* (pp. 116-131). Springer Berlin Heidelberg.

[12] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175-186). ACM.

[13] Shardanand, U., & Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 210-217). ACM Press/Addison-Wesley Publishing Co.

[14] Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, *40*(3), 66-72.

[15] Lin, W., Alvarez, S. A., & Ruiz, C. (2002). Efficient adaptive-support association rule mining for recommender systems. *Data mining and knowledge discovery*, *6*(1), 83-105.

[16] Castro-Schez, J. J., Miguel, R., Vallejo, D., & López-López, L. M. (2011). A highly adaptive recommender system based on fuzzy logic for B2C e-commerce portals. *Expert Systems with Applications*, *38*(3), 2441-2454.

[17] Lee, W. P., Liu, C. H., & Lu, C. C. (2002). Intelligent agent-based systems for personalized recommendations in Internet commerce. *Expert Systems with Applications*, *22*(4), 275-284.

[18] Srikumar, K., & Bhasker, B. (2005). Personalised recommendations in e-commerce. *International journal of electronic business*, *3*(1), 4-27.

[19] Ujjin, S., & Bentley, P. J. (2003, April). Particle swarm optimization recommender system. In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE* (pp. 124-131). IEEE.

[20] Burke, R. (2007). Hybrid web recommender systems. In *The adaptive web* (pp. 377-408). Springer Berlin Heidelberg.

[21] Mahmood, T., & Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia* (pp. 73-82). ACM.

[22] Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, *40*(3), 56-58.

[23] Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to recommender systems handbook* (pp. 1-35). Springer US.

[24] Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999, July). Combining collaborative filtering with personal agents for better recommendations. In *AAAI/IAAI* (pp. 439-446).

[25] O'Toole, R. 2013. Pedagogical strategies and technologies for peer assessment in Massively Open Online Courses (MOOCs*)*. *University of Warwick.* http://wrap.warwick.ac.uk/54602/7/WRAP_O%27toole _ROToole.

[26] Fu, X., Budzik, J., & Hammond, K. J. (2000). Mining navigation history for recommendation. In *Proceedings of the 5th international conference on Intelligent user interfaces* (pp. 106-112). ACM.

[27] Sarwar, B. M., & Adviser-Riedl, J. T. (2001). *Sparsity, scalability, and distribution in recommender systems*. University of Minnesota.

[28] Webb, G. I. (2000, August). Efficient search for association rules. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 99-107). ACM.

[29] Agarwal, R. C., Aggarwal, C. C., & Prasad, V. V. V. (2000). Depth first generation of long patterns. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 108-118). ACM.

[30] Bridge, D., Göker, M. H., McGinty, L., & Smyth, B. (2005). Case-based recommender systems. *The Knowledge Engineering Review*, *20*(03), 315-320.

[31] Ricci, F., Cavada, D., Mirzadeh, N., & Venturini, A. (2006). Case-based travel recommendations. *Destination Recommendation Systems: Behavioural Foundations and Applications*, 67-93.

[32] SapientNitro, (2012). Recommender Systems. Available at: http://www.techgig.com/expert-speak/Recommender-Systems-243 , (Accessed 21/01/2015).

[33] Basilico, J., & Hofmann, T. (2004, July). Unifying collaborative and content-based filtering. In *Proceedings of the twenty-first international conference on Machine learning* (p. 9). ACM.

[34] Wang, J., De Vries, A. P., & Reinders, M. J. (2006, August). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 501-508). ACM.

[35] Rocha, L. M. (1999). Talkmine and the adaptive recommendation project. In *Proceedings of the fourth ACM conference on Digital libraries* (pp. 242-243). ACM.

[36] Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999, August). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 230-237). ACM.