Attribute-based recommender system for learning resource by learner preference tree

Mojtaba Salehi
Faculty of Engineering, Tarbiat Modares University, Tehran, Iran
mojtaba.salehi@modares.ac.ir

Isa Nakhai Kmalabadi
Faculty of Engineering, Tarbiat Modares University, Tehran, Iran
nakhai@modares.ac.ir

***Abstract*— In recent years, with growth of online learning technology, a huge amount of e-learning resources have been generated in various media formats. This growth has caused difficulty of locating appropriate learning resources to learners. A personalized recommendation is an enabling mechanism to overcome information overload occurred in the new learning environments and deliver suitable learner resources to learners. Since users express their opinions based on some specific attributes of items, this paper considers contextual information including attributes of learning resources and rating of learner simultaneously to address some problem such as sparsity and cold start problem and also improve the quality on recommendations. Learning Tree (LT) is introduced that can model the interest of learners based on attributes of learning resources in multidimensional space using learner historical accessed resources. Then, using a new similarity measure between learners, recommendations are generated. The experimental results show that our proposed method outperforms current algorithms and alleviates problems such as cold-start and sparsity.**

Keywords- collaborative filtering, e-learning, sparsity, personalized recommender

I. Introduction

With the rapidly growth of learning resources, either offline or online in educational organizations at recent years, it is quite difficult to find suitable learning resources based on learner's preference. Typical e-learning environments that can be accessed by mobile, such as Moodle and Blackboard include course content delivery tools, synchronous and asynchronous conferencing systems, Forums, quiz modules, sharing resources, white boards and etc. [1,2]. Recommender systems are used in these environments to personalize and solve information overload. This recommendation could be an on-line activity such as doing an exercise, reading posted messages on a conferencing system, or running an on-line simulation, or could be simply a web resource [3]. One of the most important applications of recommender systems in learning environments is resources recommendation. Recommender systems help learners find the appropriate learning resources in which they would need to learn.

Many recommendation approaches have been developed in recent years. Content-based filtering and collaborative filtering (CF) are two popular types [4]. Both types of systems have inherent strengths and weaknesses, where content-based approaches directly exploit the product information, and the collaboration filtering approaches utilize specific user rating information. In the e-learning environment, learning resources are in a variety of multimedia formats including text, hypertext, image, video, audio, slides, etc. In this case, the multimedia nature makes it difficult to calculate similarity between content of two items. Therefore, in this sense, users' preference information is a good indication for recommendation. Therefore, CF is more suitable in e-learning systems since it is not necessary to analyze the content of the candidate items [5, 6].

One drawback in existing CF algorithms is that they cannot take into account attributes of item and user efficiently. Most of researches tried that only improve the accuracy of recommendation without considering satisfaction degree of user. While the recommender system algorithms try to address information overload and personalization problem, they ignore main problem in recommender systems that is user satisfaction specially for learning resource recommendation. One way to improve user satisfaction is incorporating attributes of item and user in recommendation process. Most researches don't consider contextual information such as attributes of learning resources that can address some problem such as sparsity and cold start problem and also improve the quality on recommendations.

Therefore, to improve quality and accuracy of recommendations in learning environment, this research takes into account multidimensional attributes of learning resource to model multi-preference of learner for collaborative based recommendation.

Learner preference tree (LT) is introduced that can model the interest of learners based on attributes of learning resources in multidimensional space. Then, a new similarity measure between learners is introduced to produce recommendations. The main contribution of this paper is improving the quality of recommendation and addressing sparsity problem using incorporating attributes of learning resource in the recommendation process. Using this recommender system, tutors can improve the performance of the teaching process and learners can find their suitable online resources.

Rest of this paper is organized as follows: In Literature survey section, the previous related works on m-learning and e-learning resource recommender systems is discussed.

Methodology Section introduces the overall system framework and describes the proposed mechanism step by step. Experiment section applies the proposed algorithm for a datasets to evaluate and analyze the performance. Finally, Conclusion section provides the concluding remarks.

II. Literature Review

Web-based education has undergone rapid development in recent years. With growth of many online learning systems and digitalization a lot of conventional learning resources, finding appropriate learning resources for each user has been a main problem. Those learning resources are created by individuals all over the world and are thus highly heterogeneous and dynamic. Recommender systems that are a typical example of personalization systems have been used for personalization and recommendation in learning environment specially resource recommendation. By using resource recommender systems in learning environments, we can address two problems, personalization and information overload. In this situation, recommender system offers which learning objects should learners study next [7], or offers learning objects in order to contribute to the learners' progress towards particular goals [8].

Generally, recommender systems can be divided into three categories [9]: CBF, CF, and Hybrid approaches. Content-based recommendation systems use data about the requested items and the information regarding only the active user [10]. These methods, also known as search based or item-based methods, treat the recommendation problem as a search for related items. As an example in learning environment, Khribi et al. [11] used learners' recent navigation histories and similarities and dissimilarities among user preferences and also among the contents of the learning resources for online automatic recommendations.

Most of researchers used collaborative filtering for recommendation [12]. Collaborative filtering was proposed to automate the process of ''word-of-mouth'' [13] by leveraging likeminded users' opinions. Collaborative filtering methods are completely independent of the intrinsic properties of the items being rated or recommended. Some of used techniques in this area are: user-based collaborative filtering [14], Item-based collaborative filtering [15], cluster-based collaborative filtering [16], Dimension reduction based collaborative filtering [17], Horting Graph-theoretic collaborative filtering [18]. In addition, to produce the accurate and effective recommendations and ensure the real-time requirement of the system, researchers proposed several different algorithms. The data mining techniques use the gathered information about the learner behavior, such as navigation history, to produce recommendations. Clustering was proposed by Hammouda and Kamel [19] to group learning documents based on their topics and similarities.

In education, CF holds promise not only for the purposes of helping learners and educators find useful resources, but also as a means of bringing together people with similar interests and beliefs, and possibly as an aid to the learning process itself [20].

Hybrid filtering combines collaborative and content based approach. Combining several recommendation strategies can be expected to provide better results than either strategy alone [21].

Some researches combine attributes (features) of items or users with historical rating to get better recommendations. This research tries to combine attributes of learning resource and learner in the unified model. Basu et al. [22] presented a method exploited both user ratings and content feature in recommending movies. Claypool and Gokhale [23] introduced a simple linear combination of recommendation scores from different recommenders. Burke [24] reviewed some of main approaches in these hybrid recommender systems. In summary, in order to improve the learning resource recommendation efficiency and solve some problems such as sparsity, this research develops a unified model for combining multi-dimensional attributes of resources and learner's rating information.

III. Methodology

Fig. 1 shows the framework of the proposed recommender system. At first, attributes of learning resource are extracted and weighted by experts. For learners' modeling, server usage logs of learners are collected in the certain period. Then, using this information and rating information, learner preference tree (LPT) is built for each learner. A new similarity measure between learners that can take into account information of learning trees is defined. Finally ratings are predicted for each learner and top N-recommendations are generated.

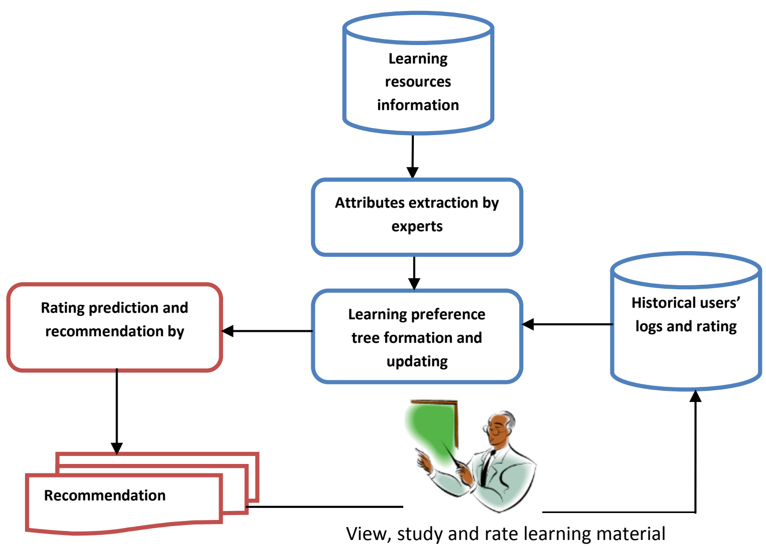

Fig. 1. System framework of the proposed recommender system

A. Attribute extraction and mateial modelling

Modeling of learner's preferences and computing the relevant resource degree between massive candidate resources and target learner is the most important task of recommendation system. We can categorize learning resources according to their attributes such as subject or different domain which each resource belongs to, for example literature, mathematics and computer science. In addition, since number of learner's accessed resources that have certain attributes indicates the importance of these attributes for the learner; it can be considered as base for weighting of attributes for the learner. Therefore, in order to consider learner's preference accurately, the multi-attribute of learning resources should be taken into account. Therefore, the resource attributes' description model can be defined as a multi-attribute vector $R =< AK_1, AK_2,..., AK_m)$ where AK_t denotes the *t*-th dimensional attribute's name of resource.

A multidimensional attribute-based framework is introduced that involves attributes of resource in the recommendation process, but selection of appropriate attributes may vary in the different systems. System developer can use Learning Object Metadata (LOM) to select suitable attributes. We select four attributes including: subject, secondary subject, education type (Bachelor Degree (B.D.), Master Degree (M.D.), PhD Degree (PhD.D.)) and author of resource. Based on this description model, the attributes of a certain resource R_j can be defined as $RA_j =< AK_1, AK_2,...AK_m >$ where AK_t denotes the *t*-th dimension attribute's keyword of resource R_j.

The central element of all recommender systems is the user model that contains knowledge about the individual preferences which determine his or her behavior in a complex environment of web-based system. According to the attention-degree of a learner to each attribute of resource, we can model interests of the learner. The attention-degree of learners is inferred by learner rating. In this paper, Learner preference tree (LPT) is introduced to combine multi-attributes of accessed resources and learner's rating information for making a multidimensional information model of learner's preference.

B. Learner preference tree

We model learners as follows: Learner preference tree has (*m*+1)-level in which *m* denotes the number of attribute of R. In this tree, the leaf node which represents an accessed resource of U_i is defined as LT_{leaf} = {RID, RR}, where RID denotes accessed resource ID of learner U_i, RR denotes the rating of U_i to certain resource (the scope is 1–5 in this paper). The non-leaf node can be defined as $LT_{nonleaf}$ = {*KA*}, where KA is the keyword of the level-th attribute of R. A learner preference tree that has four levels is shown in Fig. 2.

C. Updating strategy

In this tree, each accessed resource corresponds to a unique path from root to relevant leaf node, and the keywords of all nodes located in this path correspond to the relevant keywords of R_j 's attributes. In addition, system can update LPT using the following strategy:

Search the keywords of the latest accessed resource attributes in LPT from the upper row to the bottom. If the keyword of *i*-th attribute cannot be matched, the $m-i+1$ nodes with latter $m-i+1$ attributes of resource will be created.

D. Rating prediction

User based similarity is used to generate recommendation. As a logical assumption, two learners with

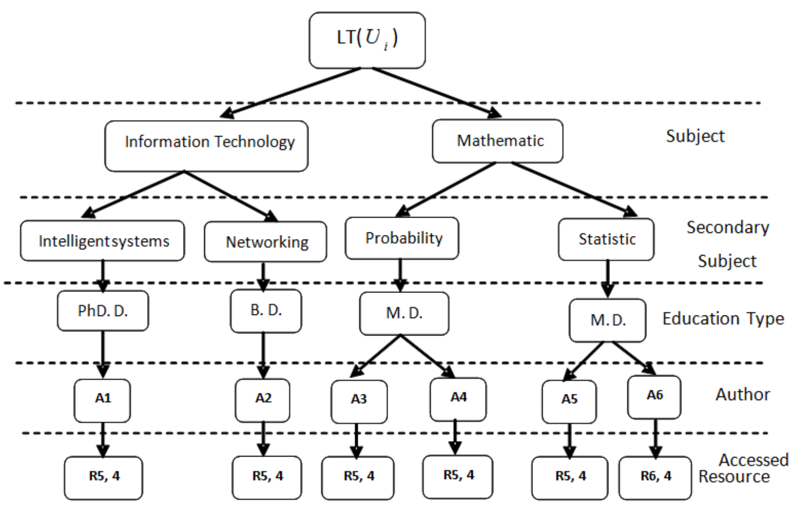

Fig. 2. Learner preference tree sample

similar attribute keywords in their LPT can be considered as similar neighbors. Based on this assumption, we can solve sparsity problem. For defining similarity degree, three rules must be considered:

(1) The more similar attributes of learner U_a and learner U_b 's accessed resources, the larger similarity between them.

(2) The more similar the order of accessed resources of learner U_a and learner U_b , the larger similarity between them.

(3) The more similar the rating data of learner U_a and learner U_b , the larger similarity between them.

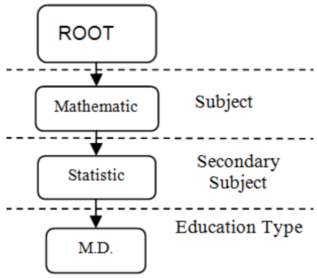

Fig. 3. Attributes Intersection Subtree (AIS) sample

Therefore, similarity degree between two learners can be calculated based on the Attributes Intersection Subtree (AIS) between two relevant LPT. *AIS* between learner U_a and U_b , $AIS(U_a,U_b)$, is defined as the maximum connected intersection between LT_a and LT_b with same node's keyword. After operating matching process, we have an *AIS* such as Fig. 3 shows.

For reflecting the similarity between the rating vectors of two learners, the learner rating based similarity can be applied to overcome sparsity rating problem. Therefore, inspired from Pearson similarity degree between learners by attribute based, $sim(U_a,U_b)$ can be computed as follows

$$sim(U_a,U_b)=\frac{\sum_{i\in L}\left|(RR_{ai}-\overline{RR}_a).(RR_{bi}-\overline{RR}_b)\right|}{\sqrt{\sum_{i\in L}(RR_{ai}-\overline{RR}_a)^2}.\sqrt{\sum_{i\in L}(RR_{bi}-\overline{RR}_b)^2}} \quad (1)$$

In Equation (1) L indicates the leaf nodes set of $AIS(U_a,U_b)$. $\overline{RR_a}$, $\overline{RR_b}$ indicate the mean value of U_a and U_b's rating data respectively. In must be noted, In the calculation of $sim(U_a,U_b)$ that computes the similarity between *RR* value of nodes on LT_a and LT_b which correspond to each leaf node on $AIS(U_a,U_b)$, does not need to have the identical accessed resources between two learners. By this definition of similarity, we can overcome sparsity rating problem.

Now we can predict rating of learning resource *i* by U_a, $P(U_a,i)$, using attribute based method. $P(U_a,i)$ is gained by the rating of U_a neighborhood, $N(U_a)$, that have rated *i* before. The computation formula is as the follows:

$$P(U_a,i)=\overline{R}_{U_a}+\frac{\sum_{j\in N(U_a)} sim(U_a,U_j)\times(R_{U_j}(i)-\overline{R}_{U_j})}{\sum_{j\in N(U_a)} sim(U_a,U_j)} \quad (2)$$

Where $\overline{R}_{U_a}$ and $\overline{R}_{U_j}$ indicate rating average of learning resources rated by active learner U_a and U_j respectively and $sim(U_a,U_b)$ is the similarity between active learner U_a and U_j that is a member of $N(U_a)$.

IV. Experiments

We have conducted a set of experiments to examine the effectiveness of our proposed recommender system.

A. *Evaluation metrics and Data set*

In order to check the performance of the proposed algorithm, a real-world dataset is applied in our simulations. MACE [1] dataset that is pan-European initiative to interconnect and disseminate digital information about architecture is used for experiment. This dataset is issued from MACE project that is done from September 2006 to September 2010. This dataset contains 1148 learners and 12000 resources.

The existing studies about recommender systems have used a number of different measures for evaluating the success of a recommender system. These measures can be divided into three categories [25]. First category, Classification Accuracy Metrics, includes methods such as Receiver Operating Characteristic curves (ROC curves) and the F1 metric that determine how often a Recommender System can decide correctly whether an item is beneficial for the user. Those metrics require a binary classification of items into useful and not useful. The second category includes Predictive Accuracy Metrics, such as Mean Absolute Error, Mean Squared error and normalized mean absolute error that measure how lose the recommender's predictions are to the true user ratings. The last category of metrics, called Rank Accuracy Metrics, measure the proximity of a predicted ordering of items, as generated by a Recommender System, to the actual user ordering of the same items.

In this research, we use from first and second category. The precision and recall are used for the evaluation of recommender system used by various researchers [25]. The precision is a measure of exactness and recall is a measure of completeness. Several ways to evaluate precision and recall exists [26]. When referring to Recommender Systems the recall can be defined as follows:

$$Recall=\frac{|test\cap top-N|}{|test|} \quad (3)$$

Where $top-N$ denotes the recommendation set and $test$ denotes the test set. The precision when referring to recommender systems can be defined as follows:

$$\Pr ecision=\frac{|test\cap top-N|}{N} \quad (4)$$

Where *N* denotes number of recommendation.

To evaluate prediction quality metric, we have used the mean absolute error (MAE), a statistical accuracy metric, [28, 29] is computed as

$$MAE=\frac{\sum_{U=1}^{U=M}|r_i(U)-\hat{r}_i(U)|}{|N|} \quad (5)$$

[1] - Metadata for Architectural Contents in Europe

Where $\hat{r}_i(U)$ is the predicted rating for resource *i* by learner *U*, r_i is the learner given rating for resource *i* by learner *U*, and *M* is the total number learners.

B. Parameters setting

Number of neighborhoods is an important parameter that must be adjusted for MACE data set in proposed recommendation system. The performance of method may vary with varying number of neighborhoods. Therefore, we setup an experiment with respect to K (the number of similar neighbors) while N varies from 5 to 40 and the minimum number of rating required for test learners, *M*, is 50

As shown in Fig. 4 when K is limited in a certain value range, with the increasing of K, the precision of each algorithm is increasing. When K reaches to a certain extent, with increasing of K, the precision of the algorithm is decreasing.

The reason is that when *K* increases to a certain extent, since several dissimilar users may be denoted as similar users by collaborative-based algorithm, the corresponding recommendation accuracy will decrease. Therefore it is necessary to set a threshold in the similar user's calculation process to guarantee the quality of them. Meanwhile the resource's attributes are taken into account based on our proposed mechanism. Therefore it can find effective similar users more accurately, and then only leads to a little performance degradation.

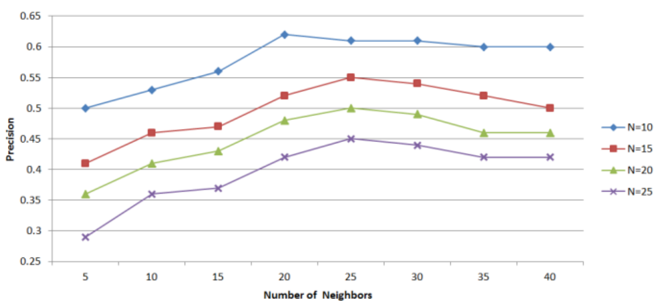

Fig. 4. Precision of algorithm with respect of K (Number of neighbors) for different number of recommendation

C. Performance comparison

In this section proposed approach is compared with some important researches in the recommender system area. In experiments, the dataset is ordered by learners' access timestamp, and then is divided into a training set and a test set.

To evaluate the sensitivity of different recommendation numbers, we compare the proposed approach and collaborative filtering based on number of recommendation for recall and precision measure with K=25, and M=50 that is presented in Fig. 5. As expected, when the number of recommendations increases, the precision drops smoothly but the recall improves gradually. The results demonstrate the effectiveness of the proposed approach.

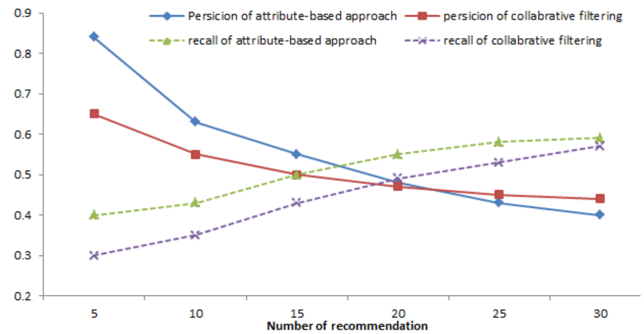

Fig. 5. Comparison of the proposed approach and CF based on number of recommendation

Table 1 presents the experimental results obtained by the proposed method, the memory-based method, Gaussian pLSA mixture method in Hofmann [27] and results published in Breese et al. [28] including Bayesian clustering (BC), Bayesian networks (BN), Correlation(CR). Since the data set will influence the results of CF algorithm, comparing of different algorithms is difficult. For the mixture pLSA, results are chosen the best results in Hofmann [27]. The results of user-based and proposed method obtained from the same data set. Comparisons were produced for K=25, and M=50. As can be seen, the proposed multi-attribute based method has better prediction accuracy of the memory-based, mixture pLSA method and other methods in terms of MAE.

Table 1: A comparison of prediction accuracy

Method	Error		
	MAE	**RMS**	**0/1 loss**
Proposed method(Attribute based)	1.106	1.356	75.2
User based	1.684	1.946	77.4
Mixture pLSA	1.190	1.568	75.7
CR [28]	1.295	-	-
BC [28]	1.407	-	-
BN [28]	1.573	-	-

The prediction accuracy improves with decreasing sparsity of data or increasing *M* for all methods, because predictions should be more reliable for learners in which a larger number of ratings is available. Fig. 6 shows the predication accuracy comparison between the user-based method and proposed method. According to Fig. 6, the relative advantage of the proposed method over the user-based method increases with decreasing *M*.

V. Conclusions

In this paper, we propose a learning resource recommendation algorithm, which utilizes multi-attributes of resources and learner rating in the unified model to have a good recommendation for learners. In the attribute based recommender, Learner preference tree (LPT) is introduced that can model the interest of learners based on attributes of learning resources in multidimensional space using historical accessed resources. The experiment results show that the proposed approach performs better than traditional collaborative filtering with sparsity increase. The main contribution of this paper is improving the quality of recommendations and addressing sparsity using problem by using multi-attributes of learners in the recommendation model.

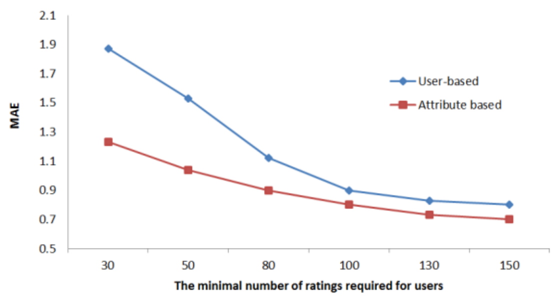

Fig. 6. MAE for the user-based method and attribute based method for different value of M

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] C. Greenhow, B. Robelia and J. E. Hughes, "Learning, Teaching, and Scholarship in a Digital Age: Web 2.0 and Classroom Research: What Path Should We Take Now? Educational Researcher," vol. 38, pp. 246-259, 2009.

[3] A. A., Kardan, S. Abbaspour, and F. Hendijanifard, "A Hybrid Recommender System for E-learning Environments Based on Concept Maps and Collaborative Tagging," In The 4th International Conference on Virtual Learning ICVL. 2009.

[4] O.R. Zaiane, "Building a recommender agent for e-learning systems. In Computers in Education," Proceedings. International Conference on. pp. 55–59, 2002.

[5] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," pp. 734–739, 2005.

[6] Y. Cai, H.-f. Leung, Q. Li, J. Tang and J. Li, "Recommendation based on object typicality", In proceedings of ACM CIKM, pp.1529–1532 2010.

[7] L. Yu, Q. Li, H. Xie and Y. Cai, "Exploring folksonomy and cooking procedures to boost cooking recipe recommendation", In proceedings of APWeb, pp. 119–130, 2011.

[8] A. Klasnja-Milicevic, B. Vesina, M. Ivanovićb and Z. Budimacb, "e-learning Personalization Based on Hybrid Recommendation Strategy and Learning Style Identification", Computers & Education, vol. 56, no. 3, pp. 885‑899, 2011.

[9] Farzan, R. & Brusilovsky, P., Social navigation support in a course recommendation system. In Adaptive Hypermedia and Adaptive Web-Based Systems. pp. 91–100, 2006.

[10] G. Adomavicius and A. Tuzhilin, "Multidimensional recommender systems: a data warehousing approach. In: 2nd international workshop on electronic commerce". Lecture Notes in Computer Science, vol. 2232, Heidelberg, Germany, pp 180–192, 2001.

[11] D. Goldberg, D. Nichols, B. M. Oki and D. Terry, "Using collaborative filtering to weave an information tapestry", Commun ACM, vol. 35, no. 12, pp. 61–70, 1992

[12] M. K. Khribi, M. Jemni, and O. Nasraoui, "Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval", In Advanced Learning Technologies. ICALT'08. Eighth IEEE International Conference on. pp. 241–245. 2008.

[13] H. Tan, J. Guo, and Y. Li, "e-learning recommendation system", In Computer Science and Software Engineering, International Conference on. pp. 430–433, 2008.

[14] U. Shardanand and P. Maes "Social information filtering: algorithms for automating 'word of mouth", In: ACM conference on human factors in computing systems (CHI'95), Denver, Colorado, USA, pp 210–217, 1995.

[15] B. Sarwar, Application of dimensionality reduction in recommender system-a case study, DTIC Document. 2000.

[16] W.S.Lee, "Collaborative learning for recommender systems", In MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-. pp. 314–321, 2001.

[17] B. Sarwar, G. Karypis, J. Konstan and J. Riedl "Analysis of recommendation algorithms for e-commerce", In Proceedings of the 2nd ACM conference on Electronic commerce. pp. 158–167, 2000.

[18] Aggarwal, C.C. et al., Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 201–212, 1999.

[19] J. L. Herlocker, "Understanding and improving automated collaborative filtering systems", Citeseer. 2000.

[20] K. Hammouda, and M. Kamel, "Collaborative document clustering", In Proceedings of the Sixth SIAM International Conference on Data Mining (SDM06). pp. 453–463, 2006.

[21] M. Recker, A . Walker and K. Lawless, "What do you recommend? Implementation and analysis of collaborative information filtering of web resources for education. Instr Sci, vol. 31, no. (4/5):pp. 299–316, 2003.

[22] M. Mahdavi, and J. Shepherd, "Enabling dynamic content caching in Web portals. In Research Issues on Data Engineering: Web Services for e-Commerce and e-Government Applications", Proceedings. 14th International Workshop on. pp. 129–136, 2004.

[23] C. Basu, Hirsh and W. Cohen, "Recommendation as classification: Using social and content-based information in recommendation", In Proceedings of the 15th national conference on artificial intelligence, pp. 714–720, 1998.

[24] M. Claypool and A. Gokhale, "Combining content-based and collaborative filters in an online newspaper", In SIGIR '99 workshop on recommender systems: Algorithms and evaluation. pp. 195–204, 1999.

[25] R.Burke, "Hybrid: Recommender systems: Survey and experiments", Journal of User Modeling and User-Adapted Interaction, vol. 12, pp. 331–370, 2002.

[26] J. Herlocker, J. Konstan and J. Riedl "An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms", Inf Retr, vol. 5, no. 4, pp. 287–310, 2002.

[27] J. L. Herlocker, J. A. Konstan, G. Terveen and J. T. Riedl, "Evaluating collaborative filtering recommender systems", ACM Transactions on Information Systems (TOIS), vol. 22, no.1, pp. 5–53, 2004.

[28] T. Hafmann, "Latent semantic models for collaborative filtering", ACM Transactions on Information Systems, vol. 22, pp. 89–115, 2004.

[29] J.S. Breese, D. Heckerman and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering", In Proceedings of the 14th conference on uncertainty in artificial intelligence pp. 43–52, 1998.