A Review Paper on Machine Learning Based Recommendation System

[1]Bhumika Bhatt, [2]Prof. Premal J Patel, [3]Prof. Hetal Gaudani
[1]M.E.C.E., [2]HOD, [2]Associate Professor
[1,2]Department of Computer Engineering, IIET, Dharmaj
[3]Department of Computer Engineering, GCET, Vallabh Vidhyanagar

***Abstract* - Recommendation system plays important role in Internet world and used in many applications. It has created the collection of many application, created global village and growth for numerous information. This paper represents the overview of Approaches and techniques generated in recommendation system. Recommendation system is categorized in three classes: Collaborative Filtering, Content based and hybrid based Approach. This paper classifies collaborative filtering in two types: Memory based and Model based Recommendation .The paper elaborates these approaches and their techniques with their limitations. This survey shows the road map for research in this area.**

***Keywords* - Recommendation, Collaborative filtering, Model based, Memory based, Content based, Hybrid.**

I. INTRODUCTION

Recommendation System is part of Daily life where people rely on knowledge for making decision of their personal interest. Recommendation system is subclass of information filtering to predict preferences to the items used by or for users. Although there are many approached developed in past but search still goes on due to it's often usage in many applications, which personalize recommendation and deals with information overload. These demands throws some challenges so different approaches like memory based, model based are used. Recommender system still requires improvement to become better system.

Recommendation system is a sharp system that provides idea about item to users that might interest them some examples are amazon.com, movies in movielens, music by last.fm. In this paper different approached with their techniques are mentioned to compare the limitation of each technique in proper manner to provide proper future recommendations.

II. BACKGROUND

A variety of approaches has been used to provide recommendation like collaborative filtering, content based and hybrid approach. Different Algorithms and approaches are there to provide recommendation that may use rating or content information; however collaborative filtering and content based method suffer from same limitations. Several researchers have tried to overcome these limitations by combining both collaborative filtering and content based method as a hybrid approach that combined ratings as well as content information. Recommendation system will always remain active search area for researchers [15].

Approaches of Recommendation System

Recommendation system is usually classified on rating estimation

- Collaborative Filtering system
- Content based system
- Hybrid system

In content-based approach, similar items to the ones the user preferred in past will be recommended to the user while in collaborative filtering, items that similar group people with similar tastes and preferences like will be recommended. In order to overcome the limitations of both approach hybrid systems are proposed that combines both approaches in some manner [15].

I. Collaborative filtering system

Collaborative filtering systems work by collecting user remark in the form of ratings for items in a given field and exploiting similarities in rating actions amongst several users in determining how to recommend an item. Collaborative filtering systems recommend an item to a user based on opinions of other users. Like, in a movie recommendation application, Collaborative filtering system tries to find other like-minded users and then recommends the movies that are most liked by them. Although there are many collaborative filtering techniques, they can be divided into two major categories [15]:

- Memory Based approaches
- Model Based approaches

1) Memory based Approach

Memory-based techniques continuously analyze all user or item data to calculate recommendations and can be classified in the following main groups: CF techniques, Content-Based (CB) techniques and hybrid techniques. CF techniques recommend items that were used by similar users in the past; they base their recommendations on social, community-driven information (e.g., user behavior like ratings or implicit histories). CB techniques recommend items similar to the ones the learners preferred in the past;

they base their recommendations on individual information and ignore the offerings from other users. Hybrid techniques combine both techniques to provide more accurate recommendations. A hybrid RS could combine CF (or social-based) techniques with CB (or information-based) techniques. If no efficient information is available to carry out CF techniques, it would switch to a CB technique [17].

Evaluating similarity between target user and training users
↓
Target user-centered formation of nearest neighborhoods
↓
Score prediction using similarity of nearest neighborhoods

Fig 1 Block Diagram of Memory Based RS [17]

The prediction process in memory-based CF contains three steps. They are similarity evaluation, generation of nearest neighborhoods and score prediction. For evaluation of the performance, the CF system considers the mean absolute error (MAE), precision and recall. The CF performance varies according to the processing method of each step[17].

A) Existing Similarity Measures

The most important first step in memory-based CF is similarity evaluation. The CF system in this step evaluates the similarity between the target user and other users for common rating items. The similarity is used as a weight for predicting the preference score. Various similarity metrics have been proposed in previous studies. These are as follows [8][10][17]:

- **Tanimoto coefficient**. It is similarity between two sets. It is a ratio of intersections. Assume that set X is {B,C, D} and set Y is {C, D, E}. The Tanimoto coefficient T of two set A and B is 0.5. This metric doesn't consider the user rating but the case of a very sparse data set is efficient[8].

$$T(X,Y) = \frac{X \cap U}{(X+Y)-(X \cap Y)} \qquad 1$$

- **Cosine similarity.** The Cosine similarity is known as the Vector similarity or Cosine coefficient. This metric assumes that common rating items of two users are two points in a vector space model, and then calculates cosϴ between the two points[10][8][18].

$$COS(U1, U2) = \frac{\sum rU1IirU2Ii}{||U1||\ ||U2||} \qquad 2$$

- **Person's Correlation**. In Equation, SU1 is the standard deviation of user U1. The Pearson Correlation measures the strength of the linear relationship between two variables. It is usually signified by *r*, and has values in the range [-1.0,1.0]. Where -1.0 is a perfect negative correlation, 0.0 is no correlation, and 1.0 is a perfect positive correlation[4][10][8][18].

$$r(U1, U2) = \frac{\sum(rU1Ii-U1)(rU2Ii-U2)}{SU1SU2} \qquad 3$$

- **Spearman's Rank Correlation.** The Spearman Rank Correlation also measures the strength of the linear relationship between two variables. Unlike the Pearson Correlation, this metric considers rank of scores. So this similarity measure has more general applicability than the Pearson Correlation, which isn't suitable outside a normalized preference range. Because the range of preference scores for CF is normalized, the Spearman Rank Correlation in the CF field shows comparable performance to the Pearson Correlation[8]

$$r(U1, U2) = \frac{6\sum(rank(rUm-rank(rUm))^2}{n(n^2-1)} \qquad 4$$

B) Formation of Nearest Neighbor

The second step after the similarity evaluation is generation of nearest neighborhoods. To improve performance, many methods have been proposed by CF researchers. The methods for selecting nearest neighborhoods include classification using K-means, a threshold for the number of common rating items and a graph algorithm. In general, it selects similar users greater than a given threshold or high rank users[8][10].

C) Prediction of Preference Score

The last step in memory-based CF is to predict the preference score of the target user for non-rating items. It predicts the preference score of non-rating items for the target user, based on the rating of nearest neighborhoods. Various methods have been proposed, and Weighted Mean is used as most general algorithm. PSU1,Ii is the predicted score of item i for U1 , and NNUi is the nearest neighbor i[8].

$$\mathrm{PSU1, Ii} = \frac{\sum \mathrm{sim(U1,NNUi)rNNUi,Ii}}{\mathrm{sim(U1,NNUi)}} \qquad 5$$

D) Performance Evaluation

In the CF system, there are two types of measure for the performance evaluation. The first type is prediction accuracy, which is evaluated by MAE. Pi is the real preference score of item i and qi is the predicted score of item i[8].

$$\mathrm{MAE} = \frac{\sum \mathrm{i-1|pi-qi|}}{\mathrm{n}} \qquad 6$$

The second type is recommendation quality, which is evaluated by precision and recall. The precision is the percentage of movies classified as higher that are higher, and recall is the percentage of higher items that were classified as higher. In addition to this, the F-measure is also used. The F-measure was proposed as means of intuitively representing the two measures and overcoming the inverse proportion of precision and recall. In equation, p is precision and r is recall[8].

$$\mathrm{F-Measure} = \frac{\mathrm{2rp}}{\mathrm{r+p}} \qquad 7$$

A user-based rating prediction can be formalized as an aggregation of the ratings that the different neighbors suggest to the target item, denoted by *fA*(*V*, *t*). These suggestions are combined by weighting the contribution of each neighbor by its similarity with respect to the active user[8].

$$ra,t = \frac{\sum V \in Nt(a) sim(A,V) fA(V,t)}{\sum V \in Nt(a) sim(A,V)} \qquad 8$$

Merits and Demerits of Memory Based Approach

User-based techniques correlate users by mining their (similar) ratings and then recommend new items that were preferred by similar users. Item-based techniques correlate the items by mining (similar) ratings and then recommend new, similar items. The main advantages of both techniques are that they use information that is provided bottom-up by user ratings, that they are domain-independent and require no content analysis and that the quality of the recommendation increases over time. CF techniques are limited by a number of disadvantages. First of all, the so-called 'cold start' problem is due to the fact that CF techniques depend on sufficient user performance from the past. Even when such systems have been running for a while, this problem emerges when new users or items are added. New users first have to give a sufficient number of ratings for items in order to get accurate recommendations based on user-based CF (new user problem)[17]. New items have to be rated by a sufficient number of users if they are to be recommended. Another disadvantage for CF techniques is the sparsity of the past user actions in a network. Since these techniques deal with community-driven information, they support well-liked tastes more strongly than unpopular tastes. The learners with an unusual taste may get less qualitative recommendations, and learners with common taste are unlikely to get unpopular items of high quality recommended. Another common problem is scalability. RSs which deal with large amounts of data, like amazon.com, have to be able to provide recommendations in real time, with the number of both the users and items exceeding millions[17].

2) Model Based Approach

In model-based CF algorithms, a theoretical model is pro-posed of user rating behavior. Rather than use the raw rating data directly in making predictions, instead the parameters of the model are estimated from the available rating data and the model is used to make predictions. Many model-based CF algorithms have been studied over the last years. For example, discusses two probabilistic models, namely, clustering and Bayesian networks. In four partitioning-based clustering algorithms are used to make predictions, leading to better scalability and accuracy in comparison to random partitioning[7].

Techniques of Model Based Approach

K-MEANS CF: *k*-means clustering is applied to identify the segments. *k*- means is a clustering method that has found wide application in data mining, statistics and machine learning. The input to *k*-means is the pair-wise distance between the items to be clustered, where the distance means the dissimilarity of the items. The number of clusters, *k* is also an input parameter. It is an iterative algorithm and starts with a random partitioning of the items into *k* clusters. Each iteration, the centroids of the clusters is computed and each item is reassigned to the cluster whose centroid is closest. The Algorithm is Described Below[7]:

Algorithm k-means clustering[7]

1. Input: R = r1
   …
   rm
2. Function kmeans(R*;* k)
3. c*i* = r*pi ;* ∀*pi* ∈ R*;* ∀*ci* ∈ C*;* ∀*i* = 1*; : : : ; k*;
4. While(*k* C*0* ¡ C *k6*= 0)
5. C*0* = C;
6. *Ci* = {*j* : *sj;i* ≥ *sj;i* ;* ∀*i** = 1*; : : : ; k*}*;* ∀*i* = 1*; : : : ; k*;

7. $ci = \frac{\Sigma rj}{|ci|}$; $\forall j \in Ci$; $\forall i = 1; : : : ; k$;
8. End While
9. return C*0*.

CLUSTER MODEL: To find customers who are similar to the user, cluster models divide the customer base into many segments and treat the task as a classification problem. The algorithm's goal is to assign the user to the segment containing the most similar customers. To find customers who are similar to the user, cluster models divide the customer base into many segments and treat the task as a classification problem[2].

The algorithm's goal is to assign the user to the segment containing the most similar customers. It then uses the purchases and ratings of the customers in the segment to generate recommendations. The segments typically are created using a clustering or other unsupervised learning algorithm, although some applications use manually determined segments. Using a similarity metric, a clustering algorithm groups the most similar customers together to form clusters or segments. Because optimal clustering over large data sets is impractical, most applications use various forms of greedy cluster generation. They then repeatedly match customers to the existing segments, usually with some provision for creating new or merging existing segments. Once the algorithm generates the segments, it computes the user's similarity to vectors that summarize each segment, then chooses the segment with the strongest similarity and classifies the user accordingly. Some algorithms classify users into multiple segments and describe the strength of each relationship. Cluster models have better online scalability and performance than collaborative filtering3 because they compare the user to a controlled number of segments rather than the entire customer base. The complex and expensive clustering computation is run offline. However, recommendation quality is low.1 Cluster models group numerous customers together in a segment, match a user to a segment, and then consider all customers in the segment similar customers for the purpose of making recommendations. Because the similar customers that the cluster models find are not the most similar customers, the recommendations they produce are less relevant[2].

BAYESIAN CLASSIFIER: Bayesian statistics that our prior beliefs are conjugate priors on $\boldsymbol{\mu}$ and Σ :

$$\Sigma \sim \text{Inv-Wishart}_{v0}(\Lambda-10\ )$$
$$\boldsymbol{\mu}|\Sigma \sim N\left\{\ \boldsymbol{\mu}0, \frac{1}{k0}\Sigma\ \right\} \qquad 9$$

where $v0$, $\Lambda0$, $\mu0$, $k0$ are hyper-parameters of the model, that is, parameters specifying our prior belief about parameters Σ and μ before observing the data. The scalar hyper parameter $v0$ describes the degrees of freedom and the matrix $\Lambda0$ describes the scale of inverse-Wishart distribution. The vector hyper-parameter $\mu0$ is the prior mean and the scalar $k0$ is the scaling of prior variance.[3][6]

$$\underbrace{P(\boldsymbol{\mu},\Sigma|\ \text{observed}\{rkl\})}_{\text{posterior belief}}\ \underbrace{P(\text{observed}\{rkl\}|\boldsymbol{\mu},\Sigma)}_{\text{likelihood}}\ \underbrace{P(\boldsymbol{\mu},\Sigma}_{\text{prior belief}}$$

The minimization can be done using the following gradient descent iterative procedure. First, It compute the gradient of the negative log posterior as follows. Let us denote elements of some index set K as $(k1, . . . , k|K|)$. Then introduce the matrix LK of size $N \times |K|$ as follows[6]:

$$\begin{cases} Lki,i = 1\ \forall i \in [1, . . . , |K|] \\ Li,j = 0 \text{ for all other elements} \end{cases} \qquad 10$$

Intuitively, if it multiply any matrix A by the matrix LK, then it just swap and arrange columns of A according to ordered set K and remove from A the columns corresponding to numbers that are not in K. For example,

$\Sigma K = L'K\Sigma LK$[6]

The estimates $\hat{\boldsymbol{\mu}}$ and $\hat{\Sigma}$ that are obtained after convergence of this algorithm are substituted into for computing ratings predictions[3][6].

III. CONTENT BASED APPROACH

Any Systems implementing a content-based recommendation approach analyze a set of documents and/or descriptions of items previously rated by a user, and build a model or profile of user interests based on the features of the objects rated by that user. The recommendation process basically consists in matching up the attributes of the user profile against the attributes of a content object. The result is a relevance judgment that represents the user's level of interest in that object. If a profile correctly reflects user preferences, it is of tremendous advantage for the effectiveness of an information access process[15].

Methods for Content Based Feature Selection[16]

1) **Wrapper methods** evaluate different subsets of features by training a model for each subset and then evaluating each subset's contribution on a validation dataset. As the number of all possible subsets is factorial in the number of features,

different heuristics are used to choose “promising" subsets (forward-selection, backward-elimination, tree-induction, etc.). Wrapper methods are independent of the prediction algorithm[16].

2) **Filter methods** are typically based on heuristic measures, such as Mutual Information or Pearson Correlation, to score features based on their information contents with respect to the prediction task. Similar to wrapper methods, filter methods are also independent of the algorithm in use. However, they do not require training many models and therefore scale well for large datasets. Yet, filter methods cannot be naturally extended to recommender systems, in which the prediction target varies and depends both on the user's history and on the item under consideration. This work proposes a framework and algorithms to address the above difficulties[16].
3) **Embedded methods** are a family of algorithms in which the feature selection is performed in the course of the training phase. Unlike wrapper methods, they are not based on cross-validation and therefore scale with the size of the data. However, since the feature selection is an inherent property of the algorithm, an embedded method is tightly coupled with the specific model: If the recommendation algorithm is replaced, features selection needs to be revisited[16].

Techniques of Content Based Approach

TF-IDF : The terms that occur frequently in one document (TF=term-frequency), but rarely in the rest of the corpus (IDF = inverse-document-frequency), are more likely to be relevant to the topic of the document. In addition, normalizing the resulting weight vectors prevent longer documents from having a better chance of retrieval. These assumptions are well exemplified by the TF-IDF function[11]:

$$\text{TF-IDF}(t_k,d_j) = \underbrace{\text{TF}(t_k,d_j)}_{\text{Tf}} \;.\; \underbrace{\log N\ n_k}_{\text{idf}} \qquad 11$$

NAÏVE BAYES: Na¨ıve Bayes is a probabilistic approach to inductive learning, and belongs to the general class of Bayesian classifiers. These approaches generate a probabilistic model based on previously observed data. The model estimates the *a posteriori* probability, $P(c|d)$, of document d belonging to class c. This estimation is based on the a priori probability, $P(c)$, the probability of observing a document in class c, $P(d|c)$, the probability of observing the document d given c, and $P(d)$, the probability of observing the instance d. Using these probabilities, the Bayes theorem is applied to calculate $P(c|d)$[11][20]:

$$P(c|d) = \frac{P(c)P(d|c)}{\mathrm{P(d)}} \qquad 12$$

Merits and Demerits of Content Based approach

The approval of the content-based recommendation paradigm has several advantages:

USER INDEPENDENCE - Content-based recommenders exploit solely ratings provided by the active user to build her own profile. Instead, collaborative filtering methods need ratings from other users in order to find the “nearest neighbors” of the active user[11].

TRANSPARENCY - Explanations on how the recommender system works can be provided by explicitly listing content features or descriptions that caused an item to occur in the list of recommendations. Those features are indicators to consult in order to decide whether to trust a recommendation[11].

NEW ITEM - Content-based recommenders are capable of recommending items not yet rated by any user. As a consequence, they do not suffer from the first-rater problem, which affects collaborative recommenders which rely solely on users’ preferences to make recommendations. Therefore, until the new item is rated by a substantial number of users, the system would not be able to recommend it[11].

content-based systems have several shortcomings:

LIMITED CONTENT ANALYSIS - Content-based techniques have a natural limit in the number and type of features that are associated, whether automatically or manually, with the objects they recommend.

OVER-SPECIALIZATION - Content-based recommenders have no inherent method for finding something unexpected. The system suggests items whose scores are high when matched against the user profile; hence the user is going to be recommended items similar to those already rated. This drawback is also called *serendipity* problem to highlight the tendency of the content-based systems to produce recommendations with a limited degree of novelty.

NEW USER - Enough ratings have to be collected before a content-based recommender system can really understand user preferences and provide accurate recommendations. Therefore, when few ratings are available, as for a new user, the system will not be able to provide reliable recommendations[11].

IV. HYBRID APPROACH

Traditional recommender system techniques such as collaborative filtering (CF), content-based, and knowledge-based filtering, each have unique strengths and limitations. For example, CF suffers from sparsity and cold start problems, while content-based approaches suffer from narrowness and require descriptions. However, a hybrid approach can use one approach to make predictions where the other fails, resulting in a more robust recommender System[1][13].

Types of Hybrid

Weighted Hybrid. In this approach, a score for each recommended item is simply the weighted sum of the recommendation scores for each source. Weights for each context source are user-configurable through interactive sliders. Automatically optimizing the set of weights for each context source is desirable, but not trivial. Empirical bootstrapping can be used to calculate an optimal weighting scheme; however, historical data is needed for this approach[13].

Mixed Hybrid. In this approach, recommendations for each source are ranked, and then the top-n are picked from each source, one recommendation at a time by alternating the sources. This approach only considers relative position in a ranked list and does not include individual recommendation scores. In cases where a recommendation is produced by multiple context sources (i.e. was previously picked from another source) the algorithm simply selects the next recommendation from the ranked list for that source[13].

Cross-Source Hybrid. This approach strongly favors recommendations that appear in more than one source. Is is believed that if a recommendation is generated from more than one context source / algorithm, i.e. by both collaborative FIltering (Facebook) and content-based recommendation (Wikipedia), then it should be considered more important. To compute a final recommendation set, the weighted hybrid approach is first applied, then each recommendation's weight is multiplied by the number of sources in which it appeared. The following equation describes the the cross-source hybrid approach:
Wreci =∑sj2S(Wreci;sj _Wsj) *| Sreci |
where jSreci j is the number of context sources recommendation i was generated by (i.e. 1, 2, or 3)[13].

How Hybrid Approach Works?
In a Movie Recommender system, the content based part of the movie recommender is based on a naive Bayesian text classification method. The classifier creates a naive Bayesian model for every user, based on the content of the movies the user has rated. The content that is used is the keywords, genres and actors of a movie and these features are assigned to an appropriate class: 1, 2, 3, 4 or 5, based on the rating for that movie. For every feature type, a separate model is created and the predictions of these models are linearly combined into one prediction[21]. The number of possible feature values of the keywords and actors would be very large if all possible values were to be used since there is a huge amount of different keywords and different actors. To be able to keep the feature vectors manageable, only the keywords are used that occur more than 20 times (8762 in total) and only the actors are used that occur more than 50 times in the data set (34956 in total).The interested posterior probability of a certain rating/class (c), given the observation of movie features (o) for a new movie. Using Bayes Theorem, this can be defined as [21]:

p(c|o) =p(c)p(o|c)/p(o) 13

Since the denominator in this equation does not depend on c and it can make use of the naive assumption that every feature fi in the observation is conditionally independent of every other feature, it can rewrite the function as[21]:

p(c|o) = p(c)n¶i=1p(fi|c) 14

where n is the number of features. In order to classify the movie, it need to find the maximum posterior probability of the five classes: classify(fi... fn) = max p(C = c)n¶i=1p(Fi = fi|C = c) So, the class/rating with the highest posterior probability for this movie is the predicted rating on which the system bases its recommendation[21].

Issue with Hybrid Approach
Reliable Integration*:* The first problem is to reflect the collaborative and content-based data when making recommendations. An easy solution is to use collaborative and content-based methods in parallel or in cascade. However, such an approach has drawbacks. Although Meta recommender systems have been proposed to select a recommender system among conventional ones on the basis of certain quality measures the disadvantages of the selected system are inherited. Moreover, the heuristics-based integration dealt with in other studies lacks a principled justification[5].
Efficient Calculation: The second problem, which has been scarcely dealt with, is to efficiently adapt a recommender system according to the increase in rating scores and users. An easy solution is to take a memory-based approach, which is originally free from this problem because the whole data is always used to make recommendations. However, these results in the late responses tried to overcome this disadvantage by using a probabilistic method in a pure collaborative filtering context. On the other hand, proposed an efficient method that incrementally trains an aspect model used for model-based collaborative filtering. To our knowledge, there are no studies on incremental adaptation of hybrid recommender systems. It need to carefully design hybrid architecture while considering whether the previous prominent methods can be applied or not[5].

V. CONCLUSION

Several recommendation systems have been anticipated are based on collaborative filtering, content based filtering and hybrid recommendation methods and so far most of them have been able to resolve the problems while providing improved recommendations. However, due to information explosion, it is required to work on this research area to explore and provide new methods that can provide recommendation in a wide range of applications while considering the quality and privacy aspects. Thus, the current recommendation system needs enhancement for present and future requirements of better recommendation qualities.

REFERENCES

[1] Alexandrin Popescu and Lyle H. Ungar, David M. Pennock and Steve Lawrence," Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments", POPESCUL ET Al,2001

[2] Greg Linden, Brent Smith, and Jeremy York,"Amazon.com Recommendations Item-to-Item Collaborative Filtering", IEEE Computer Society 2003.

[3] Jun Wang, Arjen P. de Vries, Marcel J.T. Reinders," Unifying Userbased and Itembased Collaborative Filtering Approaches by Similarity Fusion", 2006 ACM.

[4] Panagiotis Symeonidis *, Alexandros Nanopoulos, Apostolos N. Papadopoulos,Yannis Manolopoulos," Collaborative recommender systems: Combining effectiveness and efficiency", 2007 Elsevier Ltd.

[5] Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, Hiroshi G. Okuno," An Efficient Hybrid Music Recommender System Using an Incrementally TrainableProbabilistic Generative Model",IEEE 2008

[6] Akhmed Umyarov, Alexander Tuzhilin,"Improving Collaborative Filtering Recommendations Using External Data", 2008 IEEE.

[7] Zunping Cheng, Neil Hurley," Effective Diverse and Obfuscated Attacks on Model-based Recommender Systems" 2009 ACM.

[8] Hyeong-Joon Kwon, Tae-Hoon Lee, and Kwang-Seok Hong," Improved Memory-based Collaborative Filtering Using Entropy-based Similarity Measures", 2009 ACADEMY PUBLISHER.

[9] Asela Gunawardana, Christopher Meek," A Unified Approach to Building Hybrid Recommender Systems" 200X ACM

[10] Sergio Cleger-Tamayo, Juan M. Fern´andez-Luna, and Juan F. Huete, "A New Criteria for Selecting Neighborhood in Memory-Based Recommender Systems*",Springer-Verlag Berlin Heidelberg 2011.

[11] Pasquale Lops, Marco de Gemmis and Giovanni Semeraro," Content-based Recommender Systems: State of the Art and Trends", Springer 2011

[12] Yoav Bergner,Stefan Dr¨oschlery, Gerd Kortemeyer, Saif Rayyan, Daniel Seaton, and David E. Pritchard," ModelBased Collaborative Filtering Analysis of Student Response Data: MachineLearning Item Response Theory",2012.

[13] Svetlin Bostandjiev, John O'Donovan, Tobias Höllerer," TasteWeights:A Visual Interactive Hybrid Recommender System" 2012 ACM

[14] Jiankai Sun, Shuaiqiang Wang, Byron J. Gao, Jun Ma," Learning to Rank for Hybrid Recommendation", 2012 ACM

[15] Meenakshi Sharma and Sandeep Mann, "A Survey of Recommender Systems: Approaches and Limitations" ,ICAECE-2013.

[16] Royi Ronen, Noam Koenigstein, Elad Ziklik and Nir Nice," Selecting Content-Based Features for Collaborative Filtering Recommenders", ACM 2013

[17] Hendrik Drachsler, Hans G.K. Hummeland Rob Koper, "Personal recommender systems for learners in lifelong learning networks: the requirements,techniques and model".

[18] Mohammad Yahya H. Al-Shamri, Nagi H. Al-Ashwal," Fuzzy-Weighted Similarity Measures for Memory-Based Collaborative Recommender Systems", Journal of Intelligent Learning Systems and Applications, 2014.

[19] Michael J. Pazzani and Daniel Billsus," Content-based Recommendation Systems".

[20] Robin van Meteren and Maarten van Someren," Using Content-Based Filtering for Recommendation".

[21] Jaldert Rombouts, Tessa Verhoef," A simp le hybrid movie recommender system".