A Review on Recommender System

L. Anitha
V P M M Engineering College for Women,
Krishnan koil

M. K. Kavitha Devi, Ph.D
Thiagarajar college of Engineering,
Madurai

P.Anjali Devi
V P M M Engineering College for Women,
Krishnan koil

ABSTRACT
A statistics reveals that the number of people selling goods over the internet has increased by more than 100percent since 2006. Almost everyone depend on the internet for everything such as reading newspapers, magazines, books and for searching research papers, to buy latest models of all gadgets and also for entertainment like hearing songs, watching movies, and for food recipes. The internet has changed the way of living. The reason behind this is 73% time consuming and still finding exactly what we need from information available is a tedious. We expect someone to recommend the best from huge data that fulfill ones need, tastes, behavior, interest etc. The “Information Overload”- term was first coined by Alvin Toffler in his book named “Future Shock” in 1970 which is one of major issue the internet facing today. To address this issue and provide users best recommendations a System is developed called Recommender System. Recommender System applies various Data Mining methodologies to recommend efficiently for all active users based on their interest, preferences and ratings given for previous items and even based on similar users. In this paper we also analyze various issues and evaluation metrics used to measure the performance of the Recommender System.

General Terms
Various approaches, Metrics, Examples.

Keywords
Recommender System, Personalized, User ratings.

1. INTRODUCTION
Most of the Internet users have experienced the recommender system in many ways the one may be possibly “Customers Who Bought This Item Also Bought”, a list is shown of additional that are supposedly of interest. Recommender System came into existence in mid 1990s as an independent research area[1].According to Wikipedia “the recommender system form a specific type of information filtering technique that attempts to present information items(movies, music, books, news ,images) that are likely of the users interest”. If already registered user enters into the store automatically the recommender system generates a personalized[9] list of recommendations by comparing the user profile with some reference characteristic of items to predict how much the user is interested in a particular item. Item of highly interest placed in topmost of the list. Recommender System such as Ammazon.com (books, CDs, etc), Netflix, Movie Lens (Movies), Jester (Jokes), MyTripAdvisor (Travel),

Research papers (ACM, IEEE), Usenet Newsgroups etc..

Recommendation environment can be defined as[1], Let A be the set of all users and let I be the set of all possible Items that can be recommended such as movies, news, restaurants, books etc. The space V of possible items can be very large. Let f be a utility function that measures usefulness of items i to user a, i.e. $f: A \times V \rightarrow R$ where R is set of non negative ordered values within a specific range and utility function of an item noted by ratings. Then for each user $a \in A$, we want to choose such item $i' \in$ I that maximizes the user’s utility. The representation

$$\forall a \in A,\ i'a = arg\ i \in I\ max\ f(a, i) \quad (1)$$

Recommendation process are entirely based on the input (rating, user profile) provided by the visitors or users. Fig 1.1 illustrates the above mentioned process

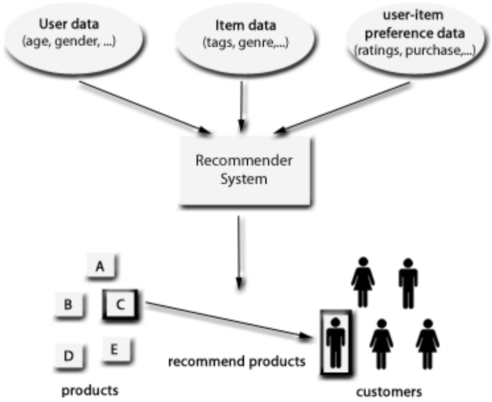

Fig 1 Recommendation Process

In next section we discuss various Recommendation approaches based on the types of data given as input to the recommender systems.

2. VARIOUS RECOMMENDATION APPROACHES:

2.1 Content based recommendation method
A content-based approach uses text-based items and analyzes the previous comments given by the user based on that the user profile is designed. The Profile is a structured representation of user interests, adopted to recommend new interesting items. The recommendation process basically consists in matching up the attributes of the user profile against the attributes of a content object (see Figure2.1). The Profiles are generated either implicitly (learning user behavior) or explicitly (through questionnaires, like/dislike).

For example to recommend news i to user a, the content-based recommender system will get the previously rated news (politics, sports, science&tech, weather) by user a and then the news with highest similarity to user preferences are recommended.

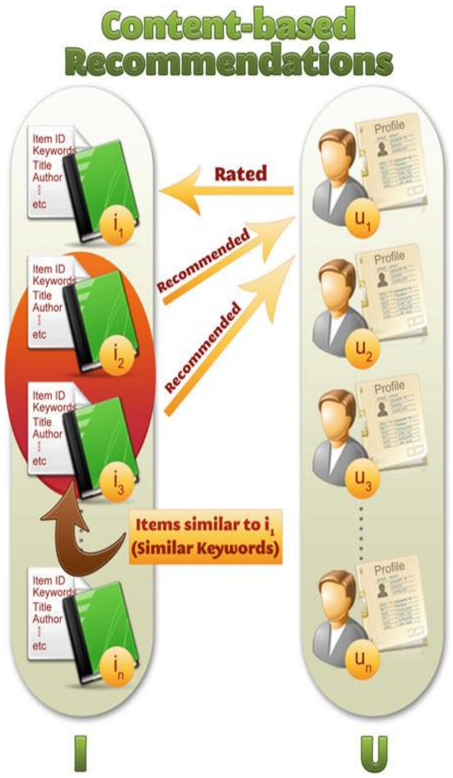

Fig 2 Content based recommendation

The content of the items is described using keyword based-information retrieval process. Most widely used similarity weighing measurements are discussed. The similar items are found by using the following measurements:

a) TF-IDF (Term frequency and Inverse Document Frequency) measure in Newsweeder (Lang, 1995), syskill and webert (pazzani et al., 1996).

$$w_{i,j} = TF_{i,j} * IDF_i$$
$$= \frac{f_{i,j}}{\sum(f_{k,j})} * \log(\frac{N}{n_i})$$

Where, N is the number of documents

n_i is how many times keyword k_i is appears in the document

$F_{i,j}$ is the number of times keyword k_i is appears in the document $_j$.

b) Probabilistic methods

Various standard (supervised) machine learning techniques can, in principle, be applied such that an intelligent system can automatically decide whether a user will be interested in a certain document. *Supervised learning* means that the algorithm relies on the existence of training data. These approaches are based on naïve bayes classification which is probabilistic one.

$$c = \frac{argmaxcj\ P\ (cj)\ P\ (d|cj)(2)}{P\ (d)}$$

Where the document with highest probability is chosen.

Limitations of content-based Recommendations:

1) Content-based recommendation only for text data types and other data types multimedia (such as images, videos, music etc) content cannot be recommended [1].

2) Uses keywords for information retrieval. In case if two items have same keyword then content based recommendation produces inaccurate results.

3) New users are not provided with good recommendation.

2.2 Collaborative Filtering recommendation method:

Collaborative Recommendation – recommend item that people with similar tastes and interests liked in the past which means the system uses the past behavior or rating of an existing user community for predicting which items the current user of the system most probably be interested in(see Fig2.2). Collaborative approach takes a matrix of given user-item ratings as input. The output will be prediction indicating to what degree user like or dislike particular item and a list of recommended items.

Ratings are the best transactional data collected by RS. Various forms of ratings are:

1) Numerical ratings based on stars (1-5) as in Amazon.com

2) Ordinal ratings such as "Strongly agree, agree, neutral, disagree, strongly disagree" users asked to show their opinion regarding the item.

3) Binary ratings simply "like or dislike" the particular item

4) Unary ratings show the behavior of the user whether the user had viewed or purchased the product.

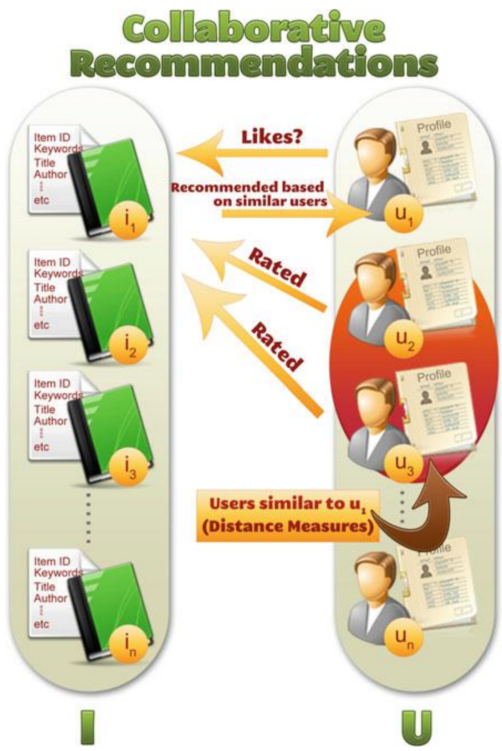

Fig 2 Collaborative Filtering Recommendation

2.2.1 User-based nearest neighbor recommendation:

In this method, given a rating DB and the ID of the active user as an input, identify other users like nearest neighbors that had similar preferences [10] to those of the active user in the past. Various similarity measures are available one among them is:

a) Pearson correlation co-efficient

$$Sim(a,b) = \frac{\Sigma p \in P(ra,p - \bar{r}a)(rb,p - \bar{r}b)}{\sqrt{\Sigma p \in P\,(ra,p - \bar{r}a)2}\sqrt{\Sigma p \in P(rb,p - \bar{r}b)2}}$$

After that Prediction is calculated,

$$\text{Pred}\,(a,b) = \bar{r}a + \frac{\Sigma b \in N sim(a,b) * (rb,p - \bar{r}b)}{\Sigma b \in N sim\,(a,b)}$$

2.2.2 Item-based nearest neighbor recommendation:

In this method the similarity between items is measured for computing the predictions.

a) Cosine similarity measure

Similarity between two items w_c and w_s are viewed with their rating vectors as

$$\cos(\vec{w}_c, \vec{w}_s) = \frac{\vec{w}_c \cdot \vec{w}_s}{||\vec{w}_c||_2 \times ||\vec{w}_s||_2}$$

After similarity measure the prediction is calculated as

$$Pred(u,p) = \frac{\Sigma wc \in \text{rated items}(u) \cos\,(wc,ws) * ru,wc}{\Sigma wc \in \text{rated items}(u) \cos\,(wc,ws)}$$

b) Matrix factorization models(SVD)

Matrix factorization models map both users and items to a joint latent factor space of dimensionality *f*, such that user-item interactions are modeled as inner products in that space. The latent space tries to explain ratings by characterizing both products and users on factors automatically inferred from user feedback.

Thus, a rating is predicted by the rule

$\hat{r}ui = \mu + bi + bu + qTi\ pu$

Limitations on collaborative recommendations:

1) Insufficient recommendations for New user :

   To solve this problem content based and collaborative recommendations are combined.

2) Insufficient rating for new item: This can also solved by hybrid approach.

2.3 Hybrid recommendation

Hybrid recommendation is a combination effective features of content- based and collaborative methods to overcome the drawbacks [1].

3. EXAMPLES

3.1 Online news Recommender System [2]

This recommender system the content-based approach and collaborative filtering approach are combined to give better recommendation to online users. The users feedback are captured in three different ways:

1) Click through the articles(f_1)
2) Ratings of articles(f_2)
3) Recommending the article to a friend by active user(f_3)

An algorithm is used to aggregate this feedback:

Case 1(No feedback available)

f_1 =0; f2 =0; f3 =0;

⇒F=0 ;(No feedback)

Case 2(Bad article)

User does not even click the main article but based on the outline, rates it as 1(low rating)

f_1 =0; f2 = (1-2)*(5/5-2) =-5/3; f3 =0;

F= $(f_1+f_2+f_3)/3$=-5/9 ;(Negative feedback)

Case 3(Best article)

Users read the articles, like it, recommend and rate it as 5.

f_1=5; f2=(5-2)*(5/5-2)=5;f_3=5;

F= $(f_1+f_2+f_3)/3$=5 ;(Maximum positive feedback)

3.2 Amazon.com Example [3]

Amazon a well known e-commerce web sites takes user interest as input and produce personalized recommendation list as output. Amazon works on item-item collaborative filtering in which similarity between items are found instead of finding similarity between users. These similar items are then added to the recommendation list. We could build a product-to-product matrix by iterating through all item pairs and computing a similarity metric for each pair. However, many product pairs have no common customers, and thus the approach is inefficient in terms of processing time and memory usage. Moreover, Amazon.com uses cosine similarity measurement. The algorithm used improves performance of the system by reducing the runtime and quick computation.

Item-item collaborative algorithm

```
Algorithm1
For each item in product catalog I1 do
  For each customer C who purchased I1 do
    For each item I2 purchase by customer C do
```

Record that a customer purchased I_1 and I_2
End for
End for
For each item I_2 do
Compute similarity between I_1 and I_2
End for
end for

3.3 last.fm

Last.fm is a social music platform that offers personalized radio streams to user the recommendation uses Audio scrobbler. Audio scrobbler builds the profiles of user such as music tracks listened by the user. User rate the music either they love the music or hate the music.

4. EVALUVATION METRICS

A set of metrics are proposed in order to evaluate the recommender system: Precision, Recall, F-measure, Fallout, ncases, Diversity, Accuracy and Friendship etc.

4.1) PRECISION (Sensitivity)

The Precision [Salton 83] a measure of exactness, determines the fraction of relevant items retrieved out of all items retrieved Precision is calculated with the following formula:

$$\text{Precision (p)} = \frac{\text{Number of successful recommendations(S)}}{\text{Number of recommendations (N)}}$$

4.2) RECALL (specificity)

The Recall [Salton 83] a measure of completeness, determines the fraction of relevant items retrieved out of all relevant items. Recall is calculated as follows

$$\text{Recall(R)} = \frac{\text{Number of recommendations (N)}}{\text{Total number of possible recommendations (T)}}$$

4.3) F-MEASURE

The F-Measure [Lewis 94] attempts to combine Precision and Recall into a single value for comparison purposes. Van Rijsbergen proposed a balanced weighting measure between precision and recall called the f-measure

$$\text{F1} = \frac{2.\text{ Prcision.recall}}{\text{Precision+recall}}$$

The result is a real value range from 0 to 1.

4.4) Ncases

The study of the average number of items (cases) contained in the user profile (case base) over time is very important, since it is desirable to reduce the size of the user profiles (solving the utility problem) while preserving or even increasing precision (while adapting the profile to the user). Certainly, the forgetting mechanism will reduce the time and the capacity needed by the algorithms to perform a recommendation.

Thus, Ncases is calculated as follows:

$$NC = \frac{\Sigma k\ i = 0|NCi|}{K}$$

Where NC_i is the number of items at the moment i, and k is the number of moments.

4.5) ACCURACY

Most widely used metrics to evaluate the system. [5]
Classified as
STATISTICAL-RECOMMENDATION ACCURACY:

1)The mean absolute error (MAE) is a measure of the deviation of recommendations from their true user-specified values.

The lower the MAE, the more accurately the recommendation engine predicts user ratings.

2) The root mean squared error (RMSE) is a measure of error biased to weigh large errors disproportionately heavier than small errors. A low RMSE indicates better accuracy.

3) Correlation is a statistical measure of agreement between two vectors of data, usually between ratings and predictions.

DECISION-SUPPORT ACCURACY:

1) ROC (Receiver operating characteristics)-
Curve used to filter the items. The curve is plotted for sensitivity and specificity. This shows the prediction range of order of items placed in Top recommendation list.

2) PRC(Precision Recall Curve) similar to ROC

Formula to calculate accuracy

$$A = \frac{\Sigma m\ i = 0|Predi - Reali\ |}{m}$$

$Pred_i$-Prediction of item i,
$Real_i$- the real evaluation of item i,
m-the number of test samples.

5. CONCLUSION:

Thus the problem of “information overload” has been solved by the Recommender system. This efficiently provides a personalized list of recommendations based on the user’s behavior and interests, so the users need not depend on someone to recommend what he has to do next. Most of the recommender system fully based on users interests (ratings made) and to recommend an item to particular user we go for finding similar users. Likewise there are item based recommendation can also be made. In this paper, Various Existing Systems are discussed and the algorithm used to produce robust output is shown. Today the collaborative recommendation algorithm out performs compared with all other approaches so we decided to implement collaborative approach for the future work, attack profile detection. Various evaluation metrics are available, Precision, Recall, ROC are mostly used.

6. REFERENCES:

[1] Adomavicius, G. & Tuzhilin, A. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, 17 (6), pp. 734–749.

[2] Manishagrawal, Maryam Karimzadehgan, Chengxiang Zhai. An Online News Recommender System for Social Networks, SIGIR-SSM, July 2009, Boston, USA.

[3] G. Linden, B. Smith, And J. York. Amazon.com Recommendations: Item-to-item collaborative Filtering. IEEE Internet Computing, 7(1):76{80, 2003.

[4] Asela Gunawardana. A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. Journal of Machine Learning Research, 10:2935{2962, 2009.

[5] B. M. Sarwar, J. A. Konstan, A. Borchers, J. Herlocker, B. Miller And J. Riedl. Using Filtering Agents to Improve Prediction Quality in the Group Lens Research Collaborative Filtering System. In Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'98), pages 345–354, 1998.

[6] Bamshadmobasher, Robin Burke,Runa Bhaumik, And Chad Williams "Towards Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness" ACM Transactions on Internet Technology,Vol. 7, No. 2, May 2007.

[7] Zhang XL, Lee TMD, Pitsilis G. Securing recommender systems against shilling attacks using social-based clustering.JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 28(4): 616{624 July 2013.

[8] Shaghayegh Sahebi., William W. Cohen " Community-Based Recommendations: a Solution to the Cold Start Problem." ACM- 2012.

[9]Zheng Siting, Hong Wenxing, Zhang Ning, Yang Fan, "Job Recommender Systems: A Survey", The 7th International Conference on Computer Science & Education (ICCSE 2012) July 14-17, 2012. Melbourne, Australia.

[10] Hyung W. Kim, Keejun Mun Y. Yi, Joonmyun Cho, Jinwoo Hong. MovieMine: Personalized Movie Content Search by Utilizing User Comments. IEEE Transactions on Consumer Electronics, vol.58, No. 4, November 2012.