Sādhanā Vol. 35, Part 5, October 2010, pp. 513–524. © Indian Academy of Sciences

Kernel based collaborative recommender system for e-purchasing

M K KAVITHA DEVI[1,*] and P VENKATESH[2]

[1]Department of Information Technology, Thiagarajar College of Engineering, Madurai 625 015
[2]Department of Electrical and Electronics Engineering, Thiagarajar College of Engineering, Madurai 625 015
e-mail: mkkdit@tce.edu; pveee@tce.edu

MS received 30 April 2008; revised 19 August 2010; accepted 13 September 2010

Abstract. Recommender system a new marketing strategy plays an important role particularly in an electronic commerce environment. Among the various recommender systems, collaborative recommender system (CRS) is widely used in a number of different applications such as recommending web pages, movies, tapes and items. CRS suffers from scalability, sparsity, and cold start problems. An intelligent integrated recommendation approach using radial basis function network (RBFN) and collaborative filtering (CF), based on Cover's theorem, is proposed in order to overcome the traditional problems of CRS. The proposed system predicts the trend by considering both likes and dislikes of the active user. The empirical evaluation results reveal that the proposed approach is more effective than other existing approaches in terms of accuracy and relevance measure of recommendations.

Keywords. Decision support system (DSS); e-purchasing; collaborative recommender system (CRS); radial basis function (RBF).

1. Introduction

The rapid expansion of the Internet has brought about a new market for trading. Electronic commerce (or) e-commerce has enabled businesses to open up their items and services to a massive client base that was once available only to the largest multinational companies. As the competition in the businesses becomes increasingly fierce, consumers are faced with a greater number of choices. This has lead to information overload problem.

Recommender systems (Goy *et al* 2007) provide one way to overcome this problem. As the name suggests, their task is to recommend items (or) products to the customer based on his/her preferences. These systems are often used by E-commerce websites as marketing tools to increase the revenue by presenting items that the customer is likely to buy. Large scale commercial applications of the recommender systems are found at many E-commerce

*For correspondence

sites, such as Amazon, CDNow, Drugstore, MovieFinder, etc. These commercial systems recommend items to potential consumers based on previous transactions and feedback. They are becoming part of the standard E-business technology that enhances E-commerce sales, increases cross-selling and builds customer loyalty.

Many recommender systems (RS) have been proposed in the literature (Adomavicius & Tuzhilin 2005). Based on the underlying technology, they are broadly categorized as content-based and collaborative filtering RS. CF approach is the most commonly-used and successful recommendation system. When predicting the potential interests of a given consumer, such an approach first identifies a set of similar consumers based on past ratings and then makes a prediction based on the observed behaviour of these similar consumers. Collaborative recommendation algorithms are grouped into two general classes: memory-based (or) heuristic-based and model-based. Memory-based algorithms essentially make rating predictions based on the entire collection of previously rated items by the users. Model-based algorithms use the collection of ratings to learn a model, which is then used to make rating predictions.

If a large number of customers' preference ratings are available for the same items, then the recommendation is accurate for the CRS. But, this may not be the case in reality because of either non-availability of a large customer pool or new items being encountered called the '*sparsity problem*'.

Many researchers have attempted to overcome the sparsity problem. Sarwar *et al* (2000), Billsus & Pazzani (1998) have used dimensionality reduction technique and singular value decomposition (SVD) to reduce the dimensionality of sparse ratings matrices. SVD is a well-known method for matrix factorization that provides the lowest rank approximations of the original matrix (Sarwar *et al* 2000). Due to consideration of only low dimensional data, this approach loses some valuable information. Another approach is to employ user profile information when calculating user similarity. That is, two users are considered similar not only if they rated the same movies similarly, but if they belong to the same demographic segment (Pazzani 1999). Another approach that explores the similarities among the users has been proposed by Huang *et al* (2004), where the sparsity problem is addressed by applying associative retrieval framework and related spreading activation algorithms to explore transitive associations among consumers through their past transactions and feedback. The other approach that explores the likes and dislikes among users using latent class model has been proposed by Cheung *et al* (2004). Instead of users, the items are clustered (Sarwar *et al* 2001) to overcome the sparsity problem.

Applying simple clustering or some statistical cluster models to the preference ratings has been demonstrated to improve the local density of the ratings and is considered to be a promising remedy for the sparsity problem. Instead of using the sparse user-item rating matrix, the matrix is initially smoothened based on average rating of the user (Xue *et al* 2005).

In this paper, the CRS is carried out by an intelligent integrated recommendation approach of RBFN and CF which use only the rating information to predict user preferences. The reason for smoothening is based on the *Cover's theorem* on the separability of patterns (1965). So, the RBFN is used to smooth the sparse data. Then CF approach provides recommendation to the customer both inside and outside the training set. Smoothening solves the traditional problems such as sparsity and cold start problems in CRS. The proposed system predicts the trend by considering both likes and dislikes of the active user. Thus, the proposed system provides high quality recommendation even for highly sparse data ($> 90\%$).

The topics are organized as follows. In section 2, CRS, their limitation and some existing systems are briefly discussed. Section 3 discusses the training phase of RBFN. In section 4, the architectural framework of the recommendation phase is proposed. Empirical results are presented in section 5. Finally, section 6 summarises the work and suggests future research directions.

2. Collaborative recommender systems (CRS)

CF is known to be the most successful recommendation techniques and is used in a number of different applications such as recommending web pages, movies, tapes and items. The CF assumes that a good way to find a certain user's interesting content is to find other people who have similar interests with him. CF methods exploit the user ratings on observed items and predict his interest on unobserved items. Despite its wide spread adoption, CRS suffers from several major limitations including scalability, sparsity and cold start (Sarwar *et al* 2000).

2.1 *Limitation in CRS*

2.1a *Scalability:* In CRS, the neighbours are identified by memory-based or model-based approach. In memory based approach the active user is compared with all the existing users; hence the computational complexity increases when the number of users increases. This problem is coined as '*scalability problem*'. In model based approach, the existing users are grouped based on their similarity. During recommendation, the similar groups are clustered together and then, only the users within the clusters are compared. So the model based approach is not affected severely by the scalability problem.

2.1b *Sparsity:* In real time E-commerce applications, huge numbers of items are presented but the user has rated or purchased very few items. This leads to sparse entries in the rating matrix. This is called '*sparsity problem*'. The sparsity level is measured as the ratio of number of zero entries to the total number of entries in the rating matrix. The consequence of the sparsity problem is lack of neighbours.

2.1c *Cold start problem:* If a user is new to the recommender system, he may not be aware of the available items and their characteristics. The user is hesitant to evaluate the items. So, he skips to rate the items in the system. Hence, the system finds it difficult to identify his neighbours. This problem is termed as '*cold start problem*'.

This paper addresses the scalability problem by clustering the users, the sparsity problem by smoothening the sparse rating matrix using RBFN and the new user cold start problem by recommending the popular items.

3. Training phase

In the model-based approach, clustering is the primary task. It is difficult to find the correlated users from a highly sparse matrix. So, the sparse user-item rating matrix is to be smoothened so that it becomes complete. The smoothening is done using RBFN.

The following details are available:

(i) A set of M users $\{u_i | i = 1, 2, \ldots, M\}$.
(ii) A set of N different items $\{p_i | i = 1, 2, \ldots, N\}$.
(iii) A rating table r_{ij} is a $[M \times N]$ matrix which contains the i^{th} user j^{th} item rating value in the rating range. The non rated items are represented by a value of zero.

3.1 *Radial basis function neural network (RBFN)*

RBFN is a family of Artificial Neural Networks which has three layers: input layer, hidden layer and output layer. The input layer contains M number of neurons, to which the user's rating vector is input. This layer is fully connected to all the neurons in the hidden layer. Each neuron in the hidden layer has activation function. The hidden layer is also fully connected to the output layer. The output layer contains M number of neurons, to which the user's smoothened rating vector is output. The output layer performs a simple summation function on users.

3.2 *Activation function*

The radial-basis function technique consists of choosing a function F

$$F(X_i) = \sum_{k=1}^{K} w_k \phi(\|X_i - C_k\|), \tag{1}$$

where w_k denote weight vector from hidden layer to output layer, X_i is the given set of points, C_k is the center of the given set of points, $\| \cdot \|$ defines norm and ϕ is the activation function. Three activation functions are attempted in this paper viz. Gaussian, multiquadratic and thin-plate spline function (Adomavicius & Tuzhilin 2005).

(i) Gaussian function:

$$\phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \text{ for some } \sigma > 0. \tag{2}$$

(ii) Multiquadratic function:

$$\phi(r) = r^{\beta} \text{ where } \beta > 0 \text{ is a positive odd number.} \tag{3}$$

(iii) Thin-plate spline function:

$$\phi(r) = r^k \log(r) \text{ where } k > 0, \tag{4}$$

where $r = \|X_i - C_k\|$, σ is the positive valued shaping parameter, β and k are positive valued parameters.

3.3 *Smoothening and clustering algorithm*

Three algorithms for smoothening the sparse rating matrix, SOM clustering, clustering, and smoothen are given below:

Algorithm 1: *Smoothen*

Input: Sparse user-item rating matrix $\langle r_{ij} \mid 1 \le i \le M \& 1 \le j \le N\rangle$, $\phi_I^{\max} \& \phi_I^{\min}$ is the maximum and minimum value of the activation function for the particular item i, r_{iI} is the rating for the item I by the user i and $\phi_{I,x}^{\max}$ is the maximum activation function value for the item I.

Output: Smothened user-item rating matrix $\langle r'_{ij} | 1 \le i \le M \& 1 \le j \le N\rangle$.

Method:

```
1. Let range = (max_ rating − min_ rating) + 1
2. Find the number of clusters $k$, such that $\lfloor \text{range}/k \rfloor \leq 3$
3. For $j = 1$ to $N$
    3.1. Partition the users into $k$ clusters using self organizing map (SOM) an unsupervised clustering technique. Let $x = r_{ij}$, where $1 \leq i \leq M$. Call function SOM($x$) (refer algorithm 2).
    3.2. Calculate the centers $C_k = \frac{\sum_{p=1}^{k1} r_{ip}}{k1}$, where $k1$ denotes number of users belong to that cluster.
    3.3. Find the Euclidean distance matrix $g_{ip} = \|r_{ip} - C_k\|$, where $1 \leq p \leq k1$. Calculate the activation function $\phi_{ip}(g)$ using (2), (3) and (4).
    3.4. Calculate the weights, using pseudo random weight function (5),
         $$\omega_i = \frac{(\phi_I^{\max} - \phi_I(r_{iI}))/(\phi_I^{\max} - \phi_I^{\min})}{\sum_{x=1}^{n} (\phi_{I,x}^{\max} - \phi_{I,x}(r_{xI}))/(\phi_x^{\max} - \phi_x^{\min})} \quad 1 \leq i \leq M. \qquad (5)$$
    3.5. Calculate $r'_{ij} = F(r_{ij})$ using equation (1).
```

Algorithm 2: *Self organizing map (SOM)*

/ SOM is a neural network used for clustering task */*

Input: Rating vector for a item $\langle x_i \mid 1 \leq i \leq M \rangle$, α_{old} is the learning rate parameter, $w_j(t)$ is the weight vector at time t, and $n1$ denotes the number of iterations.

Output: Partition k vectors.

Method:

```
1. Assign small random numbers to the initial weight $w_j = \{w_{1j}, w_{2j}, \ldots, w_{Mj}\}$, for every neuron $j$ from the output map.
2. Apply an input vector $x$.
3. Calculate the distance $d_j$ (in M-dimensional space),
       $$d_j = \sqrt{\sum (x_i - w_{ij})^2}. \qquad (6)$$
4. Calculate winner neuron as the neuron $k$ that is closest to $x$.
5. Change all the weight vectors within the neighbourhood area:
       $$\alpha_{\text{new}} = \alpha_{\text{old}} * 0{\cdot}5. \qquad (7)$$
       $$w_j(t+1) = w_j(t) + \alpha_{\text{new}} \cdot (x - w_j(t)). \qquad (8)$$
6. Repeat the steps 2 to 5 for $n1$ times.
```

Algorithm 3: *Clustering using correlation function*

Input: Smothened user-item rating matrix $\langle r'_{ij} \mid 1 \leq i \leq M \& 1 \leq j \leq N \rangle$, t are the set of items rated by both user i and j, $\overline{r'_i}$ and $\overline{r'_j}$ are the average rating of the users i and j respectively.

Output: k number of clusters [Clus$_k$ = {set_of_correlated_users}].

Method:

```
1. For $i = 1$ to $M$ /* similarity is calculated between $M$ users */
    1.1. sim_count($i$) = 0.
    1.2. For $j = 1$ to $M$.
        1.2.1. Find Pearson correlation similarity function between user $i$ and $j$
```

$$\text{sim}_{i,j} = \frac{\sum_{t \in r_i \wedge r_j} (r'_{it} - \overline{r'_i}) \cdot (r'_{jt} - \overline{r'_j})}{\sqrt{\sum_{t \in r_i \wedge r_j} (r'_{it} - \overline{r'_i})^2} \sqrt{\sum_{t \in r_i \wedge r_j} (r'_{jt} - \overline{r'_j})^2}}. \tag{9}$$

```
        1.2.2. If $\text{sim}_{i,j} = 1$ then sim_count($i$) = sim_count($i$) + 1.
    1.3. Identify top $k$ sim_count users as the cluster head for $k$ cluster.
    1.4. Assign the users to the respective clusters.
    1.5. Repeat step 2 and 3 for all the $M$ users.
```

Thus, the sparse matrix is converted to complete matrix using RBFN function. Then the users are clustered using the smoothed and actual item ratings.

4. Recommendation phase

The architectural framework of the recommendation phase is shown in figure 1. This is an online phase. In this phase the active user enters the system through the '*login session*' and gets the recommendation based on their likes and dislikes.

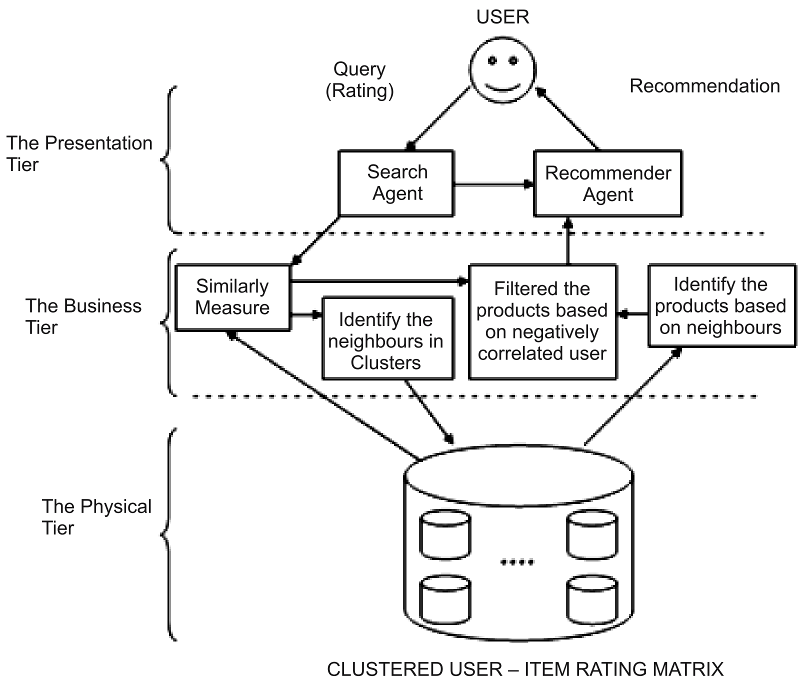

USER
Query (Rating)
Recommendation
The Presentation Tier
Search Agent
Recommender Agent
The Business Tier
Similarly Measure
Identify the neighbours in Clusters
Filtered the products based on negatively correlated user
Identify the products based on neighbours
The Physical Tier
CLUSTERED USER – ITEM RATING MATRIX

Figure 1. Architectural framework of the recommendation phase.

A new user must create his login. Once the user login the system, the user can give his ratings for some randomly selected items. Based on his ratings the recommendation is provided. The algorithmic approach is given below:

Input: Active user's rating to the items, T denotes number of items to be recommended, r_{ik} is the rating, N is the number of items, t is the set of items rated by both user i and j, $\overline{r'_i}$ and $\overline{r'_j}$ are the average rating of the users i and j respectively, $\text{csim}_{u,m}$ is the cosine similarity between the user u and m, $\overline{r'_i}$ and $\overline{r'_m}$ is the average rating of user i and m, and r'_{mi} is the rating of the item k by user i.

Output: Item recommendation for the user.

Functional specification:

1. If there are *no rated items* (cold start problem), then send T top rated items from each cluster are to the recommender agent.
2. Else, identity the positive and negative neighbouring clusters using the Pearson correlation similarity function (10).

$$\text{sim}_{i,j} = \frac{\sum_{t \in r_i \wedge r_j} (r'_{it} - \overline{r'_i}) \cdot (r'_{jt} - \overline{r'_j})}{\sqrt{\sum_{t \in r_i \wedge r_j} (r'_{it} - \overline{r'_i})^2} \sqrt{\sum_{t \in r_i \wedge r_j} (r'_{jt} - \overline{r'_j})^2}}. \tag{10}$$

3. Identify the closest positive and negative neighbors from the correlated clusters using Cosine similarity function.

$$\text{csim}_{i,j} = \frac{\sum_{k=1}^{N} r_{ik} * r_{jk}}{\sqrt{\sum_{k=1}^{N} r_{ik}^2 * \sum_{k=1}^{N} r_{jk}^2}} \tag{11}$$

4. Predict the ratings for the zero rated items in the active user rating vector using the prediction function. Choose a subset of K most similar users based on their similarity to the active user. Compute the predictions as the weighted average of deviations from the neighbour's mean

$$P_{ui} = \overline{r_i} + \frac{\sum_{m=1}^{c} (r'_{mi} - \overline{r'_m}) * \text{csim}_{u,m}}{\sum_{m=1}^{c} \text{csim}_{u,m}}. \tag{12}$$

5. Do the step 4 for the negatively correlated users.
6. Assign X as the set of recommended items predicted based on positive closest neighbours and Y as the set of recommended items predicted based on negative closest neighbours.
7. Calculate $Z = X - Y$, where X contains items based on liking and Y contains items based on disliking. Input Z to the recommender agent.
8. Recommend the *Top T* items to the active user.

5. Experimental set-up

In this section, the experimental set-up is first discussed, and then the training and recommendation phase designs are evaluated. Finally, the performance of the model is tested using the mean absolute error and relevance measure, and then it is compared with two existing models.

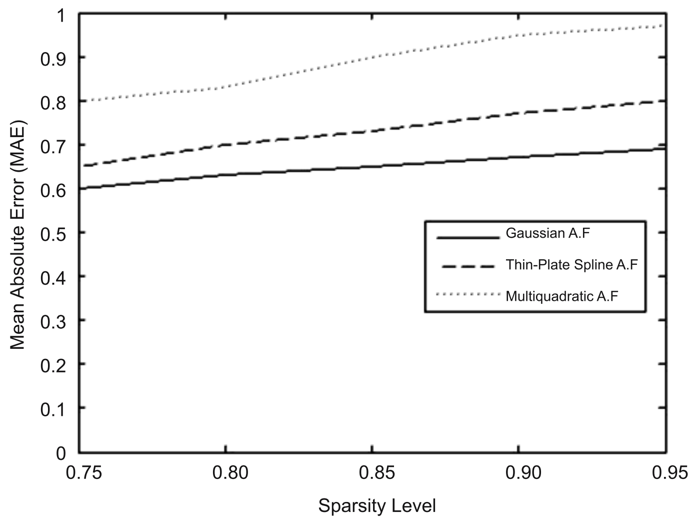


Figure 2. MAE vs. sparsity level.

5.1 *Data set*

The book crossing dataset is used to evaluate the effectiveness of the proposed CRS. The book crossing database consists of 1149780 customer preference ratings for 278858 customers and 271379 different books.

Six subsets of data are sampled from different parts of the book crossing dataset in such a way that each data subset contains 200 non-overlapping customers' preference ratings for 500 different books. The 10 subsets have various train/test ratio in the range {0·1, . . . , 0·95}.

Activation function:

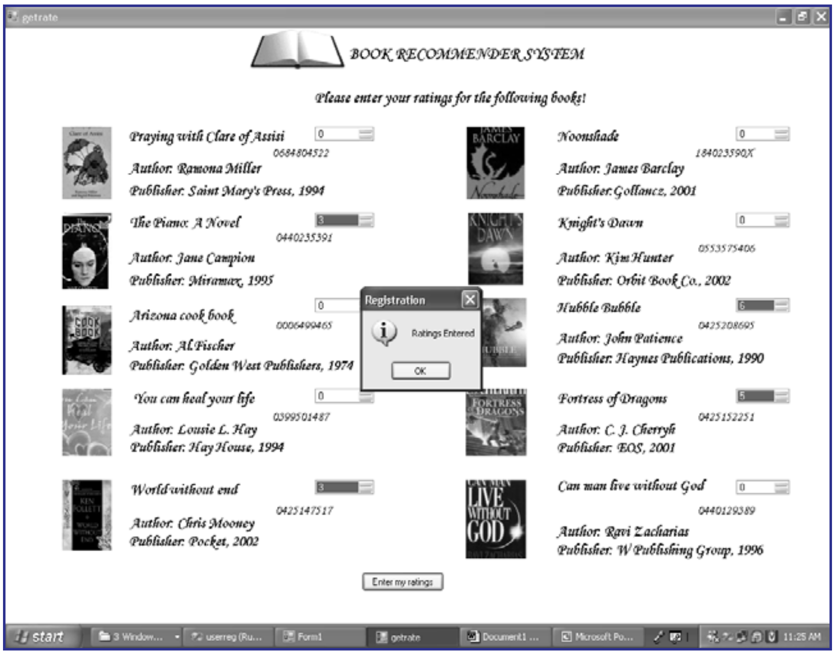
Experiments on rating matrix with various sparsity levels from 0·75 to 0·95 in a step value of 0·5 are conducted with three activation functions (3), (4) and (5). The mean absolute error (13) is obtained for various input values shown in figure 2. The observations report that Gaussian function outperforms the other two activation functions, with lowest MAE values.

The performance comparison of the proposed system with two state-of-the-art systems like low dimensional singular value decomposition CRS (Sarwar *et al* 2000), and simple CRS are provided in subsection 5·4.

5.2 *Evaluation design — recommendation phase*

The recommendation phase gives recommendation on the items to the active user. There are two activities in this phase namely:

(i) *Registration:* In this phase the active user login the system, then the user can rate as many number of randomly selected items. The respective logical design is shown in figure 3.
(ii) *Recommendation:* If the user has no rating (cold start problem), then the user gets the recommendations on the Top rated T items. Other users obtain their recommendations from the behaviour of the correlated users. The process is depicted in figure 4.


BOOK RECOMMENDER SYSTEM

Please enter your ratings for the following books!

Praying with Clare of Assisi 0 0684804522
Author: Ramona Miller
Publisher: Saint Mary's Press, 1994

The Piano: A Novel 3 0440235391
Author: Jane Campion
Publisher: Miramax, 1995

Arizona cook book 0 0006499465
Author: Al Fischer
Publisher: Golden West Publishers, 1974

You can heal your life 0 0399501487
Author: Lousie L. Hay
Publisher: Hay House, 1994

World without end 3 0425147517
Author: Chris Mooney
Publisher: Pocket, 2002

Noonshade 0 184023590X
Author: James Barclay
Publisher: Gollancz, 2001

Knight's Dawn 0 0553575406
Author: Kim Hunter
Publisher: Orbit Book Co., 2002

Hubble Bubble 6 0425208695
Author: John Patience
Publisher: Haynes Publications, 1990

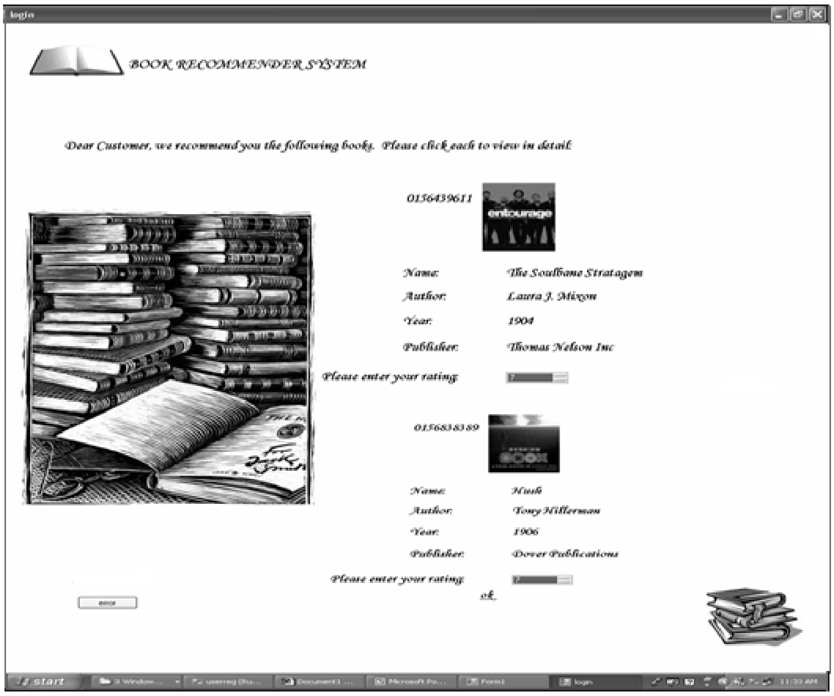
Fortress of Dragons 5 0425152251
Author: C. J. Cherryh
Publisher: EOS, 2001

Can man live without God 0 0440129389
Author: Ravi Zacharias
Publisher: W Publishing Group, 1996

Registration
Ratings Entered
OK

Enter my ratings

Figure 3. User's rating for some randomly selected books.


BOOK RECOMMENDER SYSTEM

Dear Customer, we recommend you the following books. Please click each to view in detail.

0156439611

Name: The Soulbane Stratagem
Author: Laura J. Mixon
Year: 1904
Publisher: Thomas Nelson Inc
Please enter your rating: 7

0156838389

Name: Hush
Author: Tony Hillerman
Year: 1906
Publisher: Dover Publications
Please enter your rating: 7
ok

Figure 4. Book recommendation.

5.3 *Performance evaluation*

The performance is evaluated with the following indices:

1. *Mean absolute error (MAE)*: It is a quantity used to measure how close predictions are to the eventual outcomes. The MAE is given by

$$\text{MAE} = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{|P_{ij} - r_{ij}|}{M * N}, \tag{13}$$

   where M, N, P_{ij}, r_{ij} represents number of users, number of items, predicted rating of user i for item j and true rating of user i for item.
2. *Precision and recall:* Precision is defined as the number of relevant items recommended divided by the total number of items recommended, and Recall is defined as the number of relevant items recommended divided by the total number of existing relevant items (which should have been retrieved).

$$\text{Precision} = \frac{|\{\text{relevant\_items}\} \cap \{\text{recommended\_items}\}|}{|\{\text{recommended\_items}\}|}, \tag{14}$$

$$\text{Recall} = \frac{|\{\text{relevant\_items}\} \cap \{\text{recommended\_items}\}|}{|\{\text{relevant\_items}\}|}. \tag{15}$$

3. *F-Measure:* This measure combines the precision and recall. The F-measure is given by

$$F - \text{Measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{16}$$

5.4 *Results and discussions*

5.4a *Error measure:* The deviation of the predicted rating from the actual rating of the models is measured by the mean absolute error (13) measure. The MAE of the proposed model 3 and the other two existing models 1 and 2 are shown in figure 5.

(i) *Proposed system* outperforms *Simple CR*, because there are large numbers of clusters say $L1$ before smoothen the data, but after smoothening the number of clusters is reduced to $L2$ ($L1 \gg L2$).
(ii) In SVD, only the low dimensional values are considered for predicting the active user's value, so the accuracy is low for *SVD* when comparing to *Proposed System.*

5.4b *Relevant measure:* The observations of the comparison of F-measure (figure 6) are as follows:

(i) The Precision of the *Proposed System* is 30% more than the other models.
(ii) The Recall of *Proposed System* is 75% more than the other models.
(iii) In RBF, Recall > Precision as stated in theory [2].

Experimental study is conducted for the new users, the results of the RBF smoothened outperforms while comparing to the *SVD and Simple CR.*

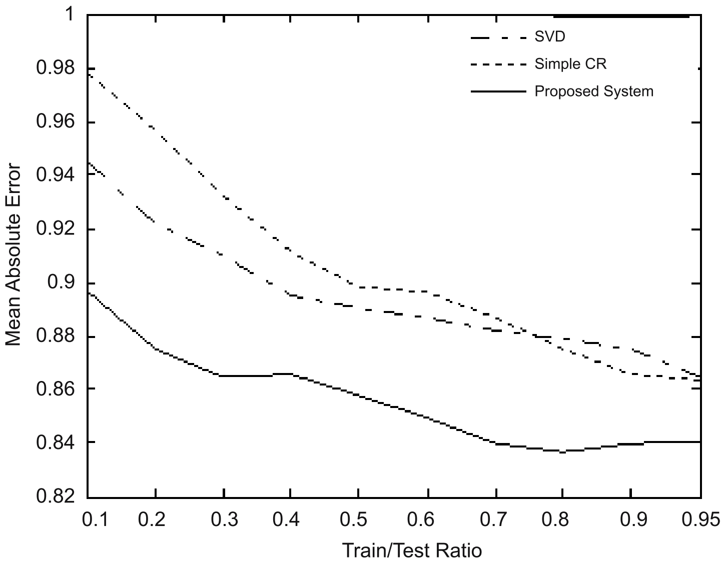


Figure 5. Comparison of models based on MAE measure.

6. Conclusion

To overcome the CRS problems, nowadays most of the recommender system uses hybrid techniques (collaborative and content based filtering) to provide recommendation.

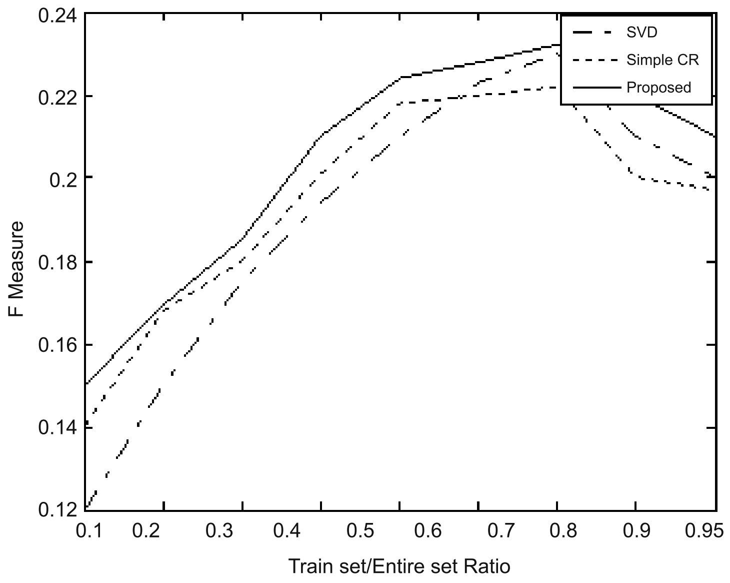


Figure 6. Comparison of the models based on F-measure.

In this paper, the sparse user-item rating matrix is used to give high quality recommendation. The existing research issues like scalability, sparsity and cold start problems in the traditional CRS are addressed in this paper.

Scalability: Traditional memory based CRS does not perform offline computation, and its online computation increases with the number of users and items. The algorithm is impossible on large data sets, unless it uses dimensionality reduction (Sarwar *et al* 2000) technique with reduced recommendation quality. In this paper, the offline (training) algorithm creates computationally expensive similar user clusters. The online (recommendation) computation is based on the number of clusters. Thus, scalability problem is curtailed to some extent.

Sparsity: CRS is purely based on the correlation between the users. That is, the recommendation for the active user is provided based on the similar liking users. In case of high sparse matrix ($> 90\%$ are zero values), it is very difficult to find the correlation between users. To overcome this difficulty, the sparse user-item matrix is converted to complete matrix using RBF methodology in the training phase. By smoothening the correlation between the users is enhanced (number of clusters before smoothening $\gg$ number of clusters after smoothening). So the RBF smoothened model offered high quality recommendation.

Cold start problem: If the user has no previous rated items, then similarity may not be established with the existing clusters. The proposed model understands that the user is a novice user, and provides a combination of top T high rated items from each cluster, as default recommendation.

So, the RBF smoothened CF model provides high quality recommendations to the users. The effectiveness of the proposed system is supported by sufficient empirical study on book data set.

References

Adomavicius G, Tuzhilin A 2005 Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowledge and Data Eng.* 17(6): 734–749

Billsus D, Pazzani M J 1998 Learning collaborative information filters. *15th International Conference on Machine Learning* 46–54

Cheung K, Tsui K, Liu J 2004 Extended latent class models for collaborative recommendation. *IEEE Trans. Systems, Man, and Cybernetics—Part A: Systems and Humans* 34(1): 143–148

Goy A, Ardissono L, Petrone G 2007 Personalization in e-commerce applications. *The Adaptive Web* (Berlin, Heidelberg: Springer-Verlag) LNCS 4321: 485–520

Huang Z, Chen H, Zeng D 2004 Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Trans. Information Systems.* 22(1): 116–142

Pazzani M J 1999 A framework for collaborative, content-based, and demographic filtering. *Artificial Intelligence Rev.* 13(5): 393–408

Sarwar B, Karypis G, Konstan J A, Riedl J T 2001 Item-based collaborative filtering recommendation algorithms. *10th International World Wide Web Conference.* 285–295

Sarwar B, Karypis G, Konstan J, Riedl J 2000 Application of dimensionality reduction in recommender systems: A case study. *ACM WebKDD Workshop*, New York

Xue G, Lin C, Yang Q, Xi W, Zeng H, Yu Y, Chen Z 2005 Scalable collaborative filtering using cluster-based smoothing. *28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil 114–121