CHAPTER 5

KNOWLEDGE BASED FUZZY LOGIC COLLABORATIVE FILTERING MODEL (KBFM)

5.1 INTRODUCTION

In this chapter Knowledge Based Fuzzy Logic Collaborative Filtering Model (KBFM) for recommender system has been presented. Knowledge based CF recommender systems generates recommendations based on the user’s personal information and preferences defined in terms of crisp rules representation. Knowledge based models recommend items based upon knowledge about user’s requirements and liking. These models have knowledge about how a particular item meets particular user requirements.

But online behaviour of users is uncertainty and impreciseness in recommender systems. To manage this challenge and issue of uncertainty of active users behavior in on line; this research proposes alternate model KBFM based on fuzzy logic for improving business by recommending right products. KBFM model is fruitfully experimented and ensured for robustness to common collaborative filtering recommender system with real world datasets namely MovieLens and Jester. In this model Fuzzy logic is used to filter the users based on fuzzy rules and Fuzzy-C-Means Clustering is used to cluster the user-item rating matrix. The related works of fuzzy based collaborative filtering technique is presented in the following section.
5.2 RELATED WORKS

In this section, review of traditional recommendation methods and fuzzy modeling methods for business intelligence recommender systems is discussed.

O.Nasraoui et al. [59] have presented the performance comparison of fuzzy inference based recommendation with collaborative filtering and nearest profile based web pages recommendation. They analysed that fuzzy based methods have outperformed in terms of coverage, speed and less memory requirement.

K. Yu, A. Schwaighofer et al. [48] in their study presented the probabilistic memory-based collaborative filtering approach to modeling uncertainties due to stochastic nature of preference. Experimental results of their approach showed that the mean precision, recall, and F1 measure of top-10 are 66%, 51%, and 57% respectively on EACHMOVIE (ratings on movies) dataset and the mean precision, recall, and F1 measure of top-10 are 40%, 47%, and 43% respectively using Jester (ratings on jokes) dataset.

Shiva Nadi et al. [74] discussed about the merits of content and collaborative filtering based techniques for generating recommendations. They proposed a fuzzy based recommender system by using collaborative behaviour of ants (FARS). This system functions in two phases namely modeling and recommendation. In the first phase, modeling is constructed in offline. In the second phase the recommendations are generated online based on the results from the first phase.
5.3 KNOWLEDGE BASED FUZZY LOGIC CF MODEL (KBFM)

In this Section, Knowledge Based Fuzzy Logic Model for collaborative filtering recommender system for providing high potential recommendations has been discussed. The proposed Knowledge Based Fuzzy Logic CF Model (KBFM) architecture is shown in Figure 5.1.
5.3.1 Methodology

This fuzzy logic based clustering approach has three phases. First phase is pre-processing in offline; the dataset is prepared for modeling by selecting relevant features using feature selection methods and dimensionality reduction to overcome the scalability problem. Second is model construction phase in offline, fuzzy and/or crisp rules are defined and applied to filter the dataset. Then clustering is applied to identify the groups from the given dataset based on the similarity of users. Third phase is intelligence phase in online, extracts business intelligence for better decision making to the businesses or active user by generating top-N recommendation list.

5.3.1.1 Pre-processing

In this phase, pre-processing has been carried out by three steps. Feature selection, dimension reduction techniques and normalization steps are applied on real word datasets to prepare the data for mining intelligence.

Feature selection (FS)

In machine learning one of the crucial problems when dealing with big data sets is the selection of the relevant features and elimination of non important features. In addressing this problem different methods of data reduction have been used and managed to eliminate the redundancy and non-important features present in the data sets. Among them Feature Selection (FS) has been shown to be a powerful approach of dealing with high dimensional data by selecting relevant features from data set at the
same time removing irrelevant and (or) redundant (highly correlated with others) features that harm the quality of the results, and therefore build a good learning model [36]. Many different feature selection techniques have been proposed in the literature. In collaborative filtering, the choice of feature selection technique is performed heuristically by the domain experts.

Normalization and Dimensionality Reduction

In this step, the user-item rating matrix is normalized between 0 and 1. Dimension reduction technique is applied to reduce the dimension from high to low. Principal Component Analysis (PCA) and decision tree induction are two major methods used for reducing the dimension. Using PCA, the optimum number of features can be identified by knee point in the percent variance against principal components plotted. Hence, in this step, dimension reduction technique especially Principal Component Analysis (PCA) is used to reduce the dimension.

5.3.1.2 Model Construction

In this second phase, the CF model is constructed using fuzzy logic. This phase has three steps, knowledge base step is used to filter of users and/or items using rules, clustering of user-item rating matrix and predicting BMC of active user using proximity measure.
Knowledge Base

In knowledge based fuzzy CF recommender systems, the filtering of users can be based on the user’s profile and preferences defined in terms of crisp and/or fuzzy rules. To express user’s knowledge based preferences and personal information the linguistic terms are used to linguistically evaluate the importance of user preferences.

Users profile and preferences are also considered as fuzzy variables. When there is uncertainty in user’s preferences and profiles, fuzzy rules are defined on membership function. For the fuzzy variable the degrees of profile matching are Too Young (TY), Young(Y), Middle (M), Old (O) and Too Old (TO). The users preferences and interest in an item consisting of Strongly Liked (SL), Liked(L), Indifferent(I), Disliked (D), and Strongly Disliked (SD) fuzzy values are defined in terms of minimum value (Min) to Maximum value (Max).

A fuzzy logic membership function defines the mapping of input to the values between 0 and 1. In practice, triangular, trapezoid, Gaussian function, S-function and exponential-like function are the most commonly used membership functions. Normally, in practice, suitable membership function's shape is assumed a priori and its parameters are determined by domain expert or using machine learning techniques [82].

Figure 5.2 illustrates the sample fuzzy logic membership function representation graph using four age groups Young, Middle, Mature and Old.
Once all the variables from dataset are identified, they are fuzzified using the membership functions. Now it is possible to define set of rules to represent input and output variables. A fuzzy rule in terms of if-then is used for defining the knowledge to manage uncertainty or impreciseness. Rules with multiple input conditions are connected using AND / OR relationships. The following rules give us to illustrate these inferences.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1: If $v_1$ is A then then $v_2$ is C</td>
<td></td>
</tr>
<tr>
<td>Rule 2: If $v_1$ is A and $v_2$ is B then $v_3$ is C</td>
<td></td>
</tr>
</tbody>
</table>

where $v_1$, $v_2$ and $v_3$ are variables/attributes and A, B and C are values. Here, is the sample input values determined into 5 ranges such as too-low, low, medium, high and too-high. The rules are evaluated based on their non-fuzzy and fuzzy behaviour. The evaluation of fuzzy rules is performed using fuzzy set operations. The operations performed on fuzzy are maximum and minimum for OR and AND operators. Once the rules are defined, these rules are evaluated to find items found in the users. The objects satisfying the crisp/fuzzy rules are filtered to reduce the dataset for mining intelligence.
When the active user is new to recommender system (Cold-start problem) the proposed model generates recommendation based on knowledge filtering of users based on demographic information and preferences of active user.

**Clustering of Utility Matrix using FCM clustering**

In this step, the utility matrix consists of ratings of user and items are grouped using Fuzzy C-Means (FCM) clustering technique to identify user’s navigation patterns. There are many clustering algorithms have been proposed in the literature. FCM is a method of clustering which allows one piece of data to belong to two or more clusters. It is also suitable for overlapped data set and comparatively better than k-means algorithm. Unlike k-means clustering where each data point belongs to only one cluster, data point is assigned membership to each cluster center leads to each data point may belong to more than one cluster center. Hence in this research, FCM clustering is used to cluster the utility matrix in terms of user-item rating matrix with user ratings dataset.

FCM clustering method (developed by Dunn in 1973 and improved by Bezdek in 1981) is frequently used in pattern recognition. This is carried out by minimizing the following objective function;

$$ j_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \| x_i - C_j \|^2, 1 \leq m < \infty $$

(5.1)

where, \( m \) is any real number greater than 1, \( u_{ij} \) is the degree of membership of \( x_i \) in the cluster \( j \), \( x_i \) is the \( i_{th} \) of d-dimensional measured data, \( C_j \) is the d-dimension center of the cluster and \( \| \cdot \| \) is any norm expression the similarity between any measured data
and the center [1][88]. FCM is an iterative method of optimizing above (5.1) function with update of membership \(u_{ij}\) and cluster centers \(c_j\) by:

\[
u_{ij} = \frac{1}{\sum_{i=1}^{N} \left( \frac{x_i - c_j}{\|x_i - c_j\|} \right)^{-\frac{2}{m-1}}} , \quad C_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}
\]

(5.2)

The iteration will stop when \(max_{ij}|u_{ij}^{(k+1)} - u_{ij}^{(k)}| < e\), where \(e\) is a termination criterion between 0 and 1, where \(k\) is number of iteration. This procedure converges to a local minimum. The steps involved in Fuzzy-C-Means clustering are given below;

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select initial centroids (C^{(0)})</td>
</tr>
<tr>
<td>2</td>
<td>compute the degree of membership in all clusters</td>
</tr>
<tr>
<td>3</td>
<td>Initialize (U= [U_{ij}]) matrix, (U^0)</td>
</tr>
<tr>
<td>4</td>
<td>Calculate centre vectors (C^{(k)} = [c_j]) with (U^{(k)})</td>
</tr>
<tr>
<td>5</td>
<td>Update (U^{(k)}, U^{(k+1)}) using degree membership.</td>
</tr>
<tr>
<td>6</td>
<td>If (max_{ij}</td>
</tr>
</tbody>
</table>

**Algorithm; FCM clustering**

**Prediction of BMC using Proximity Measure**

Prediction finds the Best Matching Cluster (BMC) of active user. It provides a numerical value, which denotes the predicted matching-score of cluster of patterns for the active customer. This predicted value is within the same scale (e.g., from 1 to 5) as the opinion values provided by active user. BMC of user’s navigation patterns of the active user is calculated by finding the similarity between the active user and cluster
centres. The similarity of the active user i.e. matching-score is calculated with other clusters. The matching-score which have users with minimum distance or more similarity is identified as best matching cluster. The proximity between two users is measured using proximity measures like Euclidean distance, Pearson Correlation and Cosine.

5.3.1.3 Recommendation

In this phase, the top-$N$ recommendations are generated from the identified best matching cluster of users. According to the sorted frequency count of items from the identified cluster of users, this returns top-$N$ items as recommendation that have not yet been purchased by the active user. A list of top-$N$ items ranked by the predicted preference is suggested to the user.

5.3.2 KBFM Algorithm

The Knowledge Based Neural Network Collaborative Filtering Model (KBFM) algorithm is presented in this section. This algorithm shows the pseudo code of the KBFM CF model.

**Input:** Training Dataset $D$; Test Dataset $TD$.

The number of clusters $k$.

$N =$ Potential number of recommendation.

**Output:** Recommendation List $\{I_1, I_2, ..., I_n\}$ of Top-$N$ items.
// Phase I: Pre-processing

Select relevant features.

Perform Normalization.

Do Dimension Reduction using PCA.

// Phase II: Model Construction

Define KB crisp rules (if any) for user preferences and/or profiles.

//Formulate the fuzzy rules (FR).

- Define the linguistic variables and terms (initialization).
- Construct the membership functions (initialization).
- Construct the rule base (initialization).

Convert crisp input data to fuzzy values using membership functions.

// Fuzzification of attributes/user preferences.

\[ \text{compute } \mu_s(a_j) \text{ where } j = 1 \text{ to } m. \]

// Fuzzification of samples/user profile features.

\[ \text{compute } \mu_s(p_j) \text{ where } j = 1 \text{ to } m. \]

Define KB fuzzy rules for user preferences and/or profiles

Apply rules and generate the resultant dataset.

If New User // Cold-Start Problem

Register and Login.

Go to Step 2 of Phase III.

Else

Do FCM Clustering of utility matrix.

//Predicting Best Matching Cluster (BMC) of active user.

For each Active user in TD do
Calculate matching score $m_s$ using proximity measure.

Select $best(m_s)$ cluster as Best Matching Cluster index $i$.

**End**

$TMAE \leftarrow$ Evaluate Matching Cluster.

**End If**

//Phase III: Recommendation

**For** each Active user in TD do

Identify items of best matching cluster users.

Calculate the frequency count of the items.

Sort the items from high to low frequency.

Select $top-N$ items.

Recommend $top-N$ items.

**End**

**Algorithm KBFM**

The time complexity of KBFM algorithm is $O(n) + O(m^2 + m)$ where $n$ is number of train data and $m$ is the number of test data. The space complexity of the algorithm is $O(1)$. 

110
5.4 EXPERIMENTAL RESULTS

In Chapter 2, a through description of various evaluation metrics utilized in related research was given. In this section detailed experimental results of the KBFM model has been presented. First the quality of clustering is analysed and validated using Silhouette Index. Second, the results of prediction evaluation using TMAE are presented. Third, recommendation quality evaluation with different parameters and metrics are presented.

5.4.1 Clustering Quality Analysis

The influence of various numbers of clusters k=2 to N/2 on clustering quality is examined. The Table 5.1 shows Mean Silhouette Index value for k=2 to N/2 number of clusters calculated using Fuzzy C Means (FCM) Clustering.

The mean Silhouette Index of FCM clustering is 0.30 and 0.39 using MovieLens and Jester dataset respectively. Since the sparsity and size of features in the Jester dataset is less than MovieLens dataset the Silhouette Index of Jester dataset is better than MovieLens dataset.
5.4.2 Prediction Accuracy Analysis

KBFM model has been implemented to identify BMC of active users. The Euclidean, Pearson Correlation and Cosine proximity measure is used to find similarity between active users and cluster of users. The model with various train data and test data is simulated for all active users to compute the Total Mean Absolute Error (TMAE) and the results are tabulated. Table 5.1 shows prediction accuracy of various proximity metrics experimented using KBFM model.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Euclidean</th>
<th>Correlation</th>
<th>Cosine</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovieLens</td>
<td>0.1382</td>
<td>0.1813</td>
<td>0.1859</td>
</tr>
<tr>
<td>Jester</td>
<td>0.1132</td>
<td>0.1217</td>
<td>0.1254</td>
</tr>
</tbody>
</table>

The results listed in the table shows that, the accuracy of Euclidean measure is better than Correlation and Cosine. When compared with Correlation and Cosine measure Euclidean measure obtained nearly 5% increase in prediction accuracy using MovieLens dataset and 1% using Jester dataset.
Figure 5.3 shows the plot for TMAE values of all active users calculated using various Proximity measures using MovieLens and Jester data set.

![Graph showing TMAE values for MovieLens and Jester datasets.]

*Fig 5.3 Prediction Accuracy TMAE.*

The above pictorial representation shows that the TMAE of Euclidean is less than the other two similarity measures using MovieLens and Jester datasets. Therefore the performance of the Euclidean is more significant than the Correlation and Cosine since it gives more accuracy in terms of TMAE.
5.4.3 Recommendation Quality Analysis

The KBFM model is tested by taking range of recommendations using MovieLens dataset from 10 to 50 in the step of 5 and 5 to 30 in the step of 5 using Jester dataset. The results are tabulated in Table 5.2 and Table 5.3.

Recommendation Quality Evaluation using MovieLens dataset

The Table 5.2 shows the Mean Recall, Precision and F1 Measures of proposed KBFM model using MovieLens dataset.

<table>
<thead>
<tr>
<th>Top-N</th>
<th>ML-dataset-1</th>
<th>ML-dataset-2</th>
<th>ML-dataset-3</th>
<th>ML-dataset-4</th>
<th>ML-dataset-5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Precision</td>
<td>F1</td>
<td>Recall</td>
<td>Precision</td>
</tr>
<tr>
<td>10</td>
<td>0.6994</td>
<td>0.7700</td>
<td>0.6917</td>
<td>0.7200</td>
<td>0.7000</td>
</tr>
<tr>
<td>15</td>
<td>0.8130</td>
<td>0.6760</td>
<td>0.6181</td>
<td>0.8872</td>
<td>0.5965</td>
</tr>
<tr>
<td>20</td>
<td>0.8778</td>
<td>0.6015</td>
<td>0.6031</td>
<td>0.9454</td>
<td>0.5181</td>
</tr>
<tr>
<td>25</td>
<td>0.9225</td>
<td>0.5408</td>
<td>0.5807</td>
<td>0.9632</td>
<td>0.4471</td>
</tr>
<tr>
<td>30</td>
<td>0.9516</td>
<td>0.4877</td>
<td>0.5528</td>
<td>0.9791</td>
<td>0.3899</td>
</tr>
<tr>
<td>35</td>
<td>0.9717</td>
<td>0.4420</td>
<td>0.5241</td>
<td>0.9760</td>
<td>0.3414</td>
</tr>
<tr>
<td>40</td>
<td>0.9823</td>
<td>0.4020</td>
<td>0.4955</td>
<td>0.9760</td>
<td>0.3002</td>
</tr>
<tr>
<td>45</td>
<td>0.9823</td>
<td>0.3651</td>
<td>0.4659</td>
<td>0.9760</td>
<td>0.2677</td>
</tr>
<tr>
<td>50</td>
<td>0.9823</td>
<td>0.3326</td>
<td>0.4378</td>
<td>0.9760</td>
<td>0.2416</td>
</tr>
</tbody>
</table>

From the table, it is observed that the maximum mean F1 measure of KBFM model is 71% and the corresponding recall and precision are 72% and 70% respectively for top-N=10.
Comparison of Recall

The experiment is repeated with different number of recommendations and calculated the mean recall, precision and F1-measure using MovieLens dataset. Figure 5.4 shows Recall measure of various number N of recommendations calculated using KBFM model. The x-axis shows the number of recommendations N range from 10 to 50 in the step of 5 and y-axis shows the mean Recall measure. This pictorial representation clearly shows that the maximum mean recall value is obtained using MovieLens dataset-4 for number of recommendations top-N=10.

![Graph showing Recall measure](image)

*Fig 5.4 Mean Recall of top-N recommendations using MovieLens dataset*
Comparison of Precision

Figure.5.5 shows mean Precision measure of various number N of recommendations calculated using this proposed KBFM method using MovieLens dataset. The x-axis shows the number of recommendations N range from 10 to 50 in the step of 5 and y-axis shows the mean Precision measure. This pictorial representation clearly shows that the maximum mean Precision value is obtained using MovieLens dataset-1 for number of recommendations top-N=10.

![Graph showing Precision vs Number of recommendations](image-url)

*Fig 5.5 Mean Precision of top-N recommendations using MovieLens dataset*
**Comparison of F1 Measure**

Figure 5.6 shows mean F1-Measure of various number N of recommendations calculated using KBFM model. The x-axis shows the number of recommendations N range from 10 to 50 in the step of 5 and y-axis shows the mean F1-Measure. This pictorial representation clearly shows that the maximum mean F1-Measure value is obtained using MovieLens dataset-2 for number of recommendations top-N=10.

![Graph showing F1-Measure vs. Number of recommendations](image)

*Fig 5.6 Mean F1-Measure of top-N recommendations using MovieLens dataset*
Recommendation Quality Evaluation using Jester dataset

The experiment is repeated with range of top-N values from 5 to 30 in the step of 5 and calculated mean Recall, Precision and F1-Measure using Jester dataset. The Table 5.3 shows the mean Recall, Precision and F1 Measures of proposed KBFM model using Jester dataset.

<table>
<thead>
<tr>
<th>Top-N</th>
<th>Jester dataset - 1</th>
<th>Jester dataset - 2</th>
<th>Jester dataset - 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Precision</td>
<td>F1</td>
</tr>
<tr>
<td>5</td>
<td>0.3391</td>
<td>0.9700</td>
<td>0.5062</td>
</tr>
<tr>
<td>10</td>
<td>0.6781</td>
<td>0.9660</td>
<td>0.8077</td>
</tr>
<tr>
<td>15</td>
<td>0.9709</td>
<td>0.9660</td>
<td>0.9727</td>
</tr>
<tr>
<td>20</td>
<td>0.9750</td>
<td>0.7593</td>
<td>0.8496</td>
</tr>
<tr>
<td>25</td>
<td>0.9780</td>
<td>0.5915</td>
<td>0.7428</td>
</tr>
<tr>
<td>30</td>
<td>0.9780</td>
<td>0.5477</td>
<td>0.7073</td>
</tr>
</tbody>
</table>

From the table, it is observed that the maximum mean F1 measure of KBFM model is 86% and the corresponding mean recall and precision are 76% and 96% respectively for top-N=10.
Comparison of Recall

Figure 5.7 shows mean Recall measure of various number N of recommendations calculated using this proposed KBFM method. The x-axis shows the number of recommendations N and y-axis shows the Recall measure. This pictorial representation clearly shows that the maximum mean Recall value is obtained using Jester-1 for number of recommendations top-N=10.

Fig 5.7 Mean Recall of top-N recommendations using Jester dataset
Comparison of Precision

Figure 5.8 shows mean Precision measure of various number N of recommendations calculated using this proposed KBFM method. The x-axis shows the number of recommendations N and y-axis shows the mean Precision measure. This pictorial representation clearly shows that the maximum mean Precision value is obtained using Jester-1 for number of recommendations top-N=10.

![Figure 5.8 Mean Precision of top-N recommendations using Jester dataset](image_url)
Comparison of F1 Measure

Figure 5.9 Shows mean F1-Measure of various number N of recommendations calculated using this proposed KBFM method. The x-axis shows the number of recommendations N and y-axis shows the mean F1-Measure. This pictorial representation clearly shows that the maximum mean Precision value is obtained using Jester-3 for number of recommendations top-N=10.

Fig 5.9 Mean F1-Measure of top-N recommendations using Jester dataset
5.4.4 Comparison with Conventional Approaches

This section explains the comparison of KBFM model with conventional k-Nearest Neighbour (k-NNBM) Approach [19][32][60][80] using MovieLens and Ant Based Recommender System (ARS) [62], k-nearest neighbour based Mean Squared Distance (MSD-CMB)[71] Jester datasets with commonly compared top-N value as 10.

Comparison using MovieLens dataset

The Table 5.4 shows the comparison of Mean F1 measure between KBFM model and conventional models using MovieLens dataset for top-N=10.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>k-NNBM</th>
<th>k-NNBM(w)</th>
<th>GAC</th>
<th>k-NNBM(PCC)</th>
<th>KBFM</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 Measure</td>
<td>0.44</td>
<td>0.53</td>
<td>0.60</td>
<td>0.66</td>
<td>0.71</td>
</tr>
</tbody>
</table>

The results listed in the table shows that, the accuracy of KBFM model is better than conventional k-NNBM, k-NNBM (w), GAC and k-NNBM (PCC) algorithms. When compared with k-NNBM, k-NNBM (w), GAC and k-NNBM (PCC) algorithms, minimum 5% increase in F1-measure is obtained in KBFM model. Therefore the performance of the KBFM is more significant than the conventional methods since it gives more accuracy in terms of F1 measure. The overall performance of KBFM is better than conventional methods.
Comparison using Jester dataset

KBFM model is compared with conventional recommender systems based on collaborative behaviour of Ants Recommender System (ARS) [62] and k-nearest neighbour based Mean Squared Distance (MSD-CMB)[71] using Jester dataset. The Table 5.5 shows the comparison of Mean F1 measure between KBFM model and conventional models using Jester dataset for $top-N=10$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ARS</th>
<th>MSD-CMB</th>
<th>KBFM</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 Measure</td>
<td>0.49</td>
<td>0.68</td>
<td>0.86</td>
</tr>
</tbody>
</table>

When compared with ARS and MSD_CMB algorithms minimum 18% increase in F1-measure is obtained in KBFM model. Therefore the performance of the KBFM is more significant than ARS and MSD_CMB methods.

5.5 SUMMARY

This research work proposes fuzzy theoretic methodology for collaborative filtering recommender system to adapt the user’s needs and improves business activities. Active user’s profile and preference features are extracted from database. It combines knowledge based user and item features and using fuzzy theories as the foundation for the effectiveness and efficiency of the KBFM model. Its advantage over other conventional recommendation method is the full personalization based on fuzzy behaviour and preferences of active users. The experimental results show that modeling uncertainty using fuzzy logic improves the performance of personalized recommender systems.